GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION

no action

Date: June 24, 1977

Project Title: Application of Linear Programming Techniques in Cut Scheduling for better Fabric Utilization

Project No: E-27-636

Project Director: Dr. Milos Konopasek

Sponsor: Camsco, Inc.

Effective Termination Date: 1/31/77

Clearance of Accounting Charges: 1/31/77

Grant/Contract Closeout Actions Remaining:

- X Final Invoice ~~and Closing Documents~~
- _ Final Fiscal Report
- _ Final Report of Inventions
- _ Govt. Property Inventory & Related Certificate
- _ Classified Material Certificate
- _ Other ____________________

Assigned to: Textile Engineering (School/Laboratory)

COPIES TO:

Project Director
Division Chief (EES)
School/Laboratory Director
Dean/Director—EES
Accounting Office
Procurement Office
✓ Security Coordinator (OCA)
Reports Coordinator (OCA)
Library, Technical Reports Section
Office of Computing Services
Director, Physical Plant
EES Information Office
Project File (OCA)
Project Code (GTRI)
Other ____________________

CA–4 (3/76)

E-27-656

B284 NK

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF TEXTILE ENGINEERING
ATLANTA, GEORGIA 30332
894-2490

APPLICATION OF LINEAR PROGRAMMING TECHNIQUES IN CUT SCHEDULING FOR BETTER FABRIC UTILIZATION

Final Report

by

Milos Konopasek
Chris Papaconstadopoulos
Howard Coff

Prepared with the support of
CAMSCO, Inc.,
Richardson, Texas 75080

February 1977

Table of Contents

Introduction

The research on "Application of Linear Programming Techniques in Cut Scheduling for Better Fabric Utilization" basically followed the schedule and yielded the results outlined in investigator's Research Proposal and sponsor's Addendum in Appendix 1 of this report.

The main results of the research are summarized in the next chapter entitled "Markamatic Cut Order Planning".

Basic general and theoretical information on the subject of the research may be found in the thesis "Cut Scheduling for Optimum Fabric Utilization in Apparel Production" in Appendix. Particular information on the results of research may be found in the sections D, P and E containing system identification, program listing and printouts from sample runs respectively.

The recommendations for further work are also given in the thesis on p. 55. It is anticipated that the first large-scale field experiments will bring about additional requirements concerning the design of user-computer dialog, report format, performance etc. Some of the features are likely to be tailored to local conditions. Two additional features will probably be required at most of the instalations: (a) interactive manipulating of requirement files; (b) round-off option for setting an integer LP solution from the first real LP solution in order to avoid a time-consuming branch & bound algorithm when solving large problems.

MARKAMATIC CUT ORDER PLANNING

The MARKAMATIC system facilitates the development and accumulating, over the period of its operation, a large number of markers for various garment styles, size distributions, fabric widths etc.

The preparation of a set of markers for every new cutting order typically includes:

a) manual selection of those of the old markers which can cover a part of the order;

b) marking the rest of the order on MARKAMATIC.

The natural limitations of human ability to process large quantities of information impedes manual retrieval and implementation of old markers in new situations. Consequently, the results of past efforts are not fully utilized, the demands on new markers may exceed available operator and system capacity, and the resulting order distribution over the set of markers is not as good as it could be.

MARKAMATIC cut order planning system developed in cooperation with Georgia Tech, computerizes a great deal of evaluation and decision making leading to the optimum proportion of (a) and (b) above. The main features of the system are as follows:

- retrieving a marker bank and selecting all the existing markers compatible with a particular cut order;
- formulation and solution of a linear programming problem which gives an optimum composition of the existing markers from the viewpoint of fabric utilization or overall operational costs;

- introduction of a concept of "dummy marker", i.e. an imaginary one-size-one garment marker with assumed efficiency: by varying this efficiency one can cover larger or smaller proportion of the new order by existing markers depending on current priorities;
- provision for cut-downs: system automatically adjusts the required size distribution according to the indicated cut-down options and the output information includes detailed instructions for cut-downs from particular markers and sizes;
- provision for complying with indicated fabric stock constraints: system reforms optimum distribution of the order considering the available quantities of the fabrics with different widths;
- flexibility in editing the input information in response to intermediate solutions: if the solution violates some obvious but difficult-to-define limitations, only a part of the solution may be accepted. The original requirements are then adjusted accordingly so that the computer can offer a modified solution, covering the remaining part of the order.
- modular design of the system allows for modification and expansion in compliance with various local conditions and constraints different from those mentioned above.

The system may be implemented and operated on existing and new MARKAMATIC installations. Also, after building an appropriate information-transfer link the system can be operated on separate dedicated or general purpose computer.

SYSTEM IDENTIFICATION

The following are the names of the programs and the supporting files of the system.

Name	Type	Name	Type
SCOR	SS	SLPM2	SS
COR	UM	LPM2	UM
SRBUL	SS	RDRC	BD
SMBUL	SS	RFILE	BD
RBULD	UM	MDIRC	BD
RMBUL	UM	MBANK	BD
SLPG	SS	ITMD1	BD
LPG	UM	ITMD2	BD
SLP8	SS	OUTIN	BD
LP8	UM	GENIN	BD

These files are on channel 1 on disc 7905 #3

Labeled as "GA TECH PROJECT"

:LI,U,1

NAME	TYPE	SCTRS	DISC	ORG	PROG LIMITS		B.P. LIMITS		ENTRY	FWAM	PB
SUBCHAN=Ø1											
%CLIB	RB	ØØØ16	TØ11	ØØØ							
%MLIB	RB	ØØØØ9	TØ11	Ø16							
FWARE	UM	ØØØØ9	TØ11	Ø25	3ØØØØ	3174Ø	Ø1ØØØ	Ø1Ø21	3ØØØØ	3174Ø	
RCOUN	RB	ØØØØ2	TØ11	Ø34							
COUN	UM	ØØØ25	TØ11	Ø36	2ØØØØ	254Ø4	ØØ572	Ø1Ø34	2ØØØØ	254Ø4	
WED	RB	ØØØØ3	TØ11	Ø61							
RMPOR	RB	ØØØØ6	TØ11	Ø64							
SMDUM	SS	ØØØØ9	TØ11	Ø7Ø							
RMDUM	RB	ØØØ1Ø	TØ11	Ø79							
MDUMP	UM	ØØØ34	TØ11	Ø89	2ØØØØ	27557	ØØ572	Ø1222	2ØØØØ	27557	
SDIRC	SS	ØØØ49	TØ12	Ø27							
RDIRC	RB	ØØØ24	TØ12	Ø76							
%WF	RB	ØØØ18	TØ13	ØØ4							
%NF	RB	ØØØ19	TØ13	Ø22							
SJBØ1	SS	ØØØØ4	TØ13	Ø41							
SBKPT	SS	ØØØØ3	TØ13	Ø45							
SZERO	SS	ØØØØ2	TØ13	Ø48							
ZERO	UM	ØØØ25	TØ13	Ø5Ø	2ØØØØ	25454	ØØ572	ØØ775	2ØØØØ	25454	
RDRC	BD	ØØ1ØØ	TØ13	Ø75							
RFILE	BD	Ø1ØØØ	TØ14	Ø79							
PDIRC	BD	ØØ1ØØ	TØ25	Ø23							
PFILE	BD	Ø1ØØØ	TØ26	Ø27							
MDIRC	BD	ØØ1ØØ	TØ36	Ø67							
MBANK	BD	Ø1ØØØ	TØ37	Ø71							
RCOR	RB	ØØØØ3	TØ48	Ø15							
SFLPA	SS	ØØØ43	TØ48	Ø18							
RFLPA	RB	ØØØ19	TØ48	Ø61							
SRLIB	SS	ØØØØ1	TØ48	Ø8Ø							
RLIB	RB	ØØØØ3	TØ48	Ø81							
SGNXC	SS	ØØØØ8	TØ48	Ø84							
RGNXC	RB	ØØØØ4	TØ48	Ø92							
CHEK	BD	ØØ3ØØ	TØ49	ØØØ							
ITMP1	BD	ØØ1ØØ	TØ52	Ø12							
ITMP2	BD	ØØ1ØØ	TØ53	Ø16							
OUTIN	BD	ØØØ5Ø	TØ54	Ø2Ø							
GENIN	BD	ØØØ5Ø	TØ54	Ø7Ø							
FILE	SS	ØØØ21	TØ55	Ø24							
LIST	SS	ØØØØ1	TØ55	Ø45							
COR	UM	ØØØ26	TØ55	Ø46	2ØØØØ	25621	ØØ572	Ø1Ø26	2ØØØØ	25621	
RMBUL	UM	ØØØ58	TØ55	Ø72	2ØØØØ	35553	ØØ572	Ø1323	2ØØØØ	35553	
RBULD	UM	ØØØ56	TØ56	Ø34	2ØØØØ	35Ø35	ØØ572	Ø1325	2ØØØØ	35Ø35	
SRBUL	SS	ØØØ12	TØ56	Ø9Ø							
SMBUL	SS	ØØØ16	TØ57	ØØ6							
SLP8	SS	ØØØ62	TØ57	Ø22							
LP8	UM	ØØØ8Ø	TØ57	Ø84	42134	65177	ØØ572	Ø1244	42134	65177	
SLPG	SS	ØØØ91	TØ58	Ø68							
LPG	UM	ØØ146	TØ59	Ø63	2Ø416	63734	ØØ572	Ø1414	2Ø416	63734	
SLPM2	SS	ØØØ62	TØ61	Ø17							
ST	SS	ØØØ61	TØ61	Ø79							
LPM2	UM	ØØ128	TØ62	Ø44	2ØØØØ	57213	ØØ572	Ø1127	2ØØØØ	57213	
SCOR	SS	ØØØ13	TØ63	Ø76							
SUBCHAN=Ø2											

```
:LI,U,1
NAME TYPE   SCTRS   DISC ORG    PROG LIMITS    B.P. LIMITS     ENTRY FWAM     PB
SUBCHAN=Ø2
%CLIB RB    ØØØ16   TØ11 ØØØ
%MLIB RB    ØØØØ9   TØ11 Ø16
FWARE UM    ØØØØ9   TØ11 Ø25    3ØØØØ 3174Ø    Ø1ØØØ Ø1Ø21     3ØØØØ 3174Ø
RCOUN RB    ØØØØ2   TØ11 Ø34
COUN  UM    ØØØ25   TØ11 Ø36    2ØØØØ 254Ø4    ØØ572 Ø1Ø34     2ØØØØ 254Ø4
WED   RB    ØØØØ3   TØ11 Ø61
RMPOR RB    ØØØØ6   TØ11 Ø64
SMDUM SS    ØØØØ9   TØ11 Ø7Ø
RMDUM RB    ØØØ1Ø   TØ11 Ø79
MDUMP UM    ØØØ34   TØ11 Ø89    2ØØØØ 27557    ØØ572 Ø1222     2ØØØØ 27557
SDIRC SS    ØØØ49   TØ12 Ø27
RDIRC RB    ØØØ24   TØ12 Ø76
%WF   RB    ØØØ18   TØ13 ØØ4
%NF   RB    ØØØ19   TØ13 Ø22
SJBØ1 SS    ØØØØ4   TØ13 Ø41
SBKPT SS    ØØØØ3   TØ13 Ø45
SZERO SS    ØØØØ2   TØ13 Ø48
ZERO  UM    ØØØ25   TØ13 Ø5Ø    2ØØØØ 25454    ØØ572 ØØ775     2ØØØØ 25454
RDRC  BD    ØØ1ØØ   TØ13 Ø75
RFILE BD    Ø1ØØØ   TØ14 Ø79
PDIRC BD    ØØ1ØØ   TØ25 Ø23
PFILE BD    Ø1ØØØ   TØ26 Ø27
MDIRC BD    ØØ1ØØ   TØ36 Ø67
MBANK BD    Ø1ØØØ   TØ37 Ø71
RCOR  RB    ØØØØ3   TØ48 Ø15
SFLPA SS    ØØØ43   TØ48 Ø18
RFLPA RB    ØØØ19   TØ48 Ø61
SRLIB SS    ØØØØ1   TØ48 Ø8Ø
RLIB  RB    ØØØØ3   TØ48 Ø81
SGNXC SS    ØØØØ8   TØ48 Ø84
RGNXC RB    ØØØØ4   TØ48 Ø92
CHEK  BD    ØØ3ØØ   TØ49 ØØØ
ITMP1 BD    ØØ1ØØ   TØ52 Ø12
ITMP2 BD    ØØ1ØØ   TØ53 Ø16
OUTIN BD    ØØØ5Ø   TØ54 Ø2Ø
GENIN BD    ØØØ5Ø   TØ54 Ø7Ø
COR   UM    ØØØ26   TØ55 Ø24    2ØØØØ 25621    ØØ572 Ø1Ø26     2ØØØØ 25621
RMBUL UM    ØØØ58   TØ55 Ø5Ø    2ØØØØ 35553    ØØ572 Ø1323     2ØØØØ 35553
RBULD UM    ØØØ56   TØ56 Ø12    2ØØØØ 35Ø35    ØØ572 Ø1325     2ØØØØ 35Ø35
SRBUL SS    ØØØ12   TØ56 Ø68
SMBUL SS    ØØØ16   TØ56 Ø8Ø
LLL   SS    ØØØØ1   TØ57 ØØØ
SLP8  SS    ØØØ62   TØ57 ØØ1
LP8   UM    ØØØ8Ø   TØ57 Ø63    42134 65177    ØØ572 Ø1244     42134 65177
SLPG  SS    ØØØ91   TØ58 Ø47
LPG   UM    ØØ146   TØ59 Ø42    2Ø416 63734    ØØ572 Ø1414     2Ø416 63734
SLPM2 SS    ØØØ62   TØ6Ø Ø92
ST    SS    ØØØ61   TØ61 Ø58
LPM2  UM    ØØ128   TØ62 Ø23    2ØØØØ 57213    ØØ572 Ø1127     2ØØØØ 57213
SCOR  SS    ØØØ13   TØ63 Ø55
@
```

PROGRAM INDEX

```
:LI,S,1,SCOR

:JO
:PU,ST,COR
:CL
:ST,S,ST,10
FTN4
      PROGRAM COR
C
C     ***********************************************
C     *               PROGRAM COR                  *
C     ***********************************************
C
C     PROGRAM COR IS THE MONITOR OF THE SYSTEM.
C     DEPENDING ON THE CONTROL COMMAND IT ROUTES CONTROL
C     TO THE ASSOCIATED MODULO.
C     THE FOLLOWING (IN THEIR LOGICAL SEQUENCE) COMMANDS
C     ARE AVAILABLE :
C
C          COMMAND                       ASSOC. MODULE (SOURCE)
C
C       1. $R[EQUEST]                   RBULD (SRBUL)
C       2. $B[UILD]                     RMBULD (SMBUL)
C       3. $M[ODEL]                     LPG (SLPG)
C       4. $E[XECUTE]                   LP8 (SLP8)
C       5. $P[RINT]                     LPM2 (SLPM2)
C
C     ***************************************************
C
C            ARRAY SIZE DECLARATION IN THE PROGRAMS
C            OF THE PACKAGE
C
C       N1  :  MAX NUMBER OF SIZES IN A REQUIREMENT FILE
C
C       N2  :  MAX NUMBER OF REGULAR MARKERS
C
C       N3  :  MAX NUMBER OF FABRIC WIDTHS
C
C       N4  :  MAX NUMBER OF REGULAR AND DUMMY MARKERS
C              N4=N1+N2
C
C       N5  :  MAX NUMBER OF CONSTRAINTS  N5=N4+NINT
C              WHERE NINT : ADDITIONAL INTEGER LP CONSTRAINTS
C              NINT APPROX= N1+N3
C
C       N6  :  MAX NUMBER OF REJECTED MARKERS AND
C              MAX NUMBER OF SIZES IN CUT-DOWN GROUPS
C
C       N7  :  MAX NUMBER OF CUT-DOWNS
C
C       N8  :  MAX NUMBER OF NON-ZERO ELEMENTS IN P,Q,IP,IQ ARRAYS
C
C     ***********************************************************
C
```

```
C
      DIMENSION ICMD(6),IPAR(5),IP1(3),IP2(3),IP3(3),IP4(3),IP5(3)
      DATA IP1/2HRB,2HUL,2HD /,IP3/2HLP,2H8 ,2H  /
      DATA ICMD/2H$R,2H$B,2H$M,2H$E,2H$P,2H$S/
      DATA IP5/2HLP,2HM2,2H  /
      DATA LOUT/1/
      DATA IP2/2HRM,2HBU,2HLD/
      DATA IP4/2HLP,2HG ,2H  /
C
C
C         CHECK INPUT PARAMETER
C
      CALL RMPAR(IPAR)
      IF(IPAR(1).EQ.1)GO TO 150
C
   90 WRITE(LOUT,100)
100   FORMAT("ENTER INPUT DEVICE #")
      READ(1,*)LIN
C
C          RETURN FROM WORKER SEGMENT
      WRITE(LOUT,110)
  110 FORMAT("  ENTER CONTROL CARD")
C
150   READ(LIN,200)JCMD
200   FORMAT(A2)
C
C                 EVALUATE THE COMMAND
      DO 1000 I=1,6
      IF(JCMD.EQ.ICMD(I))GO TO 2000
1000  CONTINUE
      WRITE(LOUT,300)
300   FORMAT("ILLEGAL COMMAND")
      GO TO 90
C
C          ROUTE CONTROL TO CORRECT WORKER PROGRAMS
C
2000  GOTO(3000,4000,5000,6000,7000,8000),I
```

```
C
C               BUILD RECQUIREMENT FILE
C
3000   CALL EXEC(10,IP1,LIN,LOUT)
C
C               BUILD MARKER BANK
C
4000   CALL EXEC(10,IP2,LIN,LOUT)
C
C                BUILD PROGLEM FILE
C
5000   CALL EXEC(10,IP4,LIN,LOUT)
C
C                  EXECUTE THE PROBLEM
C
6000   CALL EXEC(10,IP3,LIN,LOUT)
C
C                  GENERATE THE PRODUCTION REPORT
C
7000   CALL EXEC(10,IP5,LIN,LOUT)
C
8000   STOP
       END
       END$
::
:JF,ST
:PR,FTN4,2,1,99
:LU,7,0
:PR,LOADR,2,7,0,0,0
%MLIB,%CLIB,/E
:ST,P
:PU,ST
:CL
:EJ
:TY
```

**** LIST END ****

0

```
:LI,S,1,SRBUL

:JO
:PU,RBULD,ST
:CL
:ST,S,ST,10
FTN4
      PROGRAM RBULD
C
C     ***********************************************
C     *                 PROGRAM SRBUL               *
C     ***********************************************
C
C     SRBUL IS RESPONSIBLE FOR STORING THE REQUIREMENT
C     FILE AND UPDATING THE SUPPORTING DIRECTORY FILE
C     ( BY CUT-ID )
C     RFILE IS THE REQUIREMENT FILE. THE FIRST CARD (WHICH
C     GOES INTO THE DIRECTORY FILE) MUST BE THE CUT-ID WITH
C     THE FOLLOWING FORMAT : CT--<CUT-ID>
C     WHERE -- IS TWO(2) BLANKS.
C     THE SECOND CARD ( FIRST CARD IN THE REQ. FILE ) MUST
C     BE A STYLE CARD WITH FORMAT : ST<STYLE NAME>
C     THE MAX LENGTH OF THE CUT-ID OR STYLE NAME IS TEN(10)
C     CHARACTERS. THE FOLLOWING THE STYLE CARD, CARDS MUST BE
C     EITHER ...
C       A.  SIZE CARDS WITH FORMAT : SZ--<SIZE ID>,<QUANTITY>,C
C           WHERE C IS Y[ES] OR N[O] FOR THE CUT DOWNS.
C            THE DEFAULT ONE IS THE N[O]
C     OR...
C       B.  PIECE GOOD CARDS WITH FORMAT : PG--<WIDTH>,<LENGTH>
C
C     THE CARDS ARE STORED IN RFILE AS CARD IMAGES OF 20 CHARACTE
C     THE REQUIREMENT FILE MUST BE TERMINATED BY A /E.
C     REQ.FILES ARE STORED BY CUT-ID THROUGH A DIRECTORY FILE
C     ( RDRC )
C
C     ***********************************************************
C
      DIMENSION IPAR(5),ICRD(10),IRF(3),IP(3)
      DIMENSION  IBUF(128)
      DATA IP/2HCO,2HR ,2H  /
      DATA IRF/2HRF,2HIL,2HE /
      DATA LOUT/1/
C
C         GET LOGICAL UNIT OF INPUT DEVICE
C
      CALL RMPAR(IPAR)
      LIN =IPAR(1)
      NCNT=2
      ICNT=1
```

```
C
C           READ CARD IMAGE
C
   50  IF(ICRD(1).EQ.2H/E) GOTO 1020
       READ(LIN,100) ICRD
       IF(ICRD(1).EQ.2HCT)GO TO 1000
 100   FORMAT(10A2)
       DO 300 I=1,10
       IBUF(NCNT)=ICRD(I)
 300   NCNT=NCNT+1
       IF(NCNT.NE.122)GO TO 50
       CALL EXEC(15,66,IBUF,128,IRF,NSCTR)
       NSCTR=NSCTR+1
       ICNT=ICNT+1
       NCNT=2
       DO 400 I=1,128
 400   IBUF(I)=0
       GOTO 50
C
C             MAKE DIRECTORY ENTRY
C
 1000  CALL DIREC(2005B,ICRD(2),INTNO,NSCTR,ISTAT)
       CALL FLPAK(5)
       CALL DIREC(405B,ICRD(2),INTNO,NSCTR,ISTAT)
       IF(ISTAT.NE.0)GO TO 1010
       IFSTR=NSCTR
       GO TO 50
C                             OUTPUT ERROR MESSAGE
 1010  WRITE(LOUT,200) ISTAT
 200   FORMAT("BAD STATUS RFILE =",I3)
       GO TO 50
C
C            HIT TERMINATOR COMMAND, CLEAN-UP
C
 1020  IF(NCNT.EQ.2)GO TO 1030
       CALL EXEC(15,66,IBUF,128,IRF,NSCTR)
C               UPDATE FIRST SECTOR
 1025  CALL EXEC(14,66,IBUF,128,IRF,IFSTR)
       IBUF(1)=ICNT
       CALL EXEC(15,66,IBUF,128,IRF,IFSTR)
C               LOAD MONITOR
       CALL EXEC(10,IP)
 1030  ICNT=ICNT-1
       GO TO 1025
       END
       END$
::
:JF,ST
:PR,FTN4,2,1,99
:LU,7,0
:PR,LOADR,2,7
RLIB
%MLIB,%CLIB,RDIRC,RFLPA,/E
:ST,P
```

```
:LI,S,1,SMBUL

:JO
:PU,RMBUL,ST
:CL
:ST,S,ST,10
FTN4
      PROGRAM RMBUL
C
C     ***************************************************
C     *               PROGRAM SMBUL                     *
C     ***************************************************
C
C     SMBUL IS RESPONSIBLE FOR STORING THE MARKER BANK
C    AND UPDATING THE SUPPORTING DIRECTORY FILE ( BY STYLE )
C    MBANK IS THE FILE FOR THE MARKER BANK.
C    THE MARKER BANK CONSISTS OF TWO PARTS :
C     1.  PATTERN AREAS WITH FORMAT : <SIZE NAME>,<AREA>
C         EACH CARD IS STORED AS A CARD IMAGE OF 20 CHARACTERS
C         THIS SECTION MUST BE TERMINATED BY A  /E
C     2.  MARKERS  WITH FORMAT :
C          <MARKER-ID>,<WIDTH>,<LENGTH>,[K*]<SIZE NAME>,.../
C          WHERE K IS THE NUMBER OF GARMENTS OF THE SAME SIZE.
C          THE DEFAULT VALUE OF K IS ONE (1).
C         CONTINUATION IS ALLOWED ON COMMAS.
C         EACH MARKER MUST BE TERMINATED BY A SLASH (/)
C         THE LAST MARKER MUST BE TERMINATED BY A /@
C         FOLLOWED BY A  /E  TO TERMINATE THE OPERATION
C
C          THE PROGRAM READS IN CARDS OF 72 COLUMNS. THE BLANKS
C         AT THE END OF EACH CARD ARE SUPRESSED. HENCE THE
C         MARKER SECTION IS STORED AS A CONTINUOUS STRING
C
C     ***********************************************************
C
      DIMENSION IPAR(5),ICRD(36),MBF(3),IP(3),IBUF(128)
      DATA IP/2HCO,2HR ,2H  /
      DATA MBF/2HMB,2HAN,2HK /
      DATA IBLANK/2H  /
C
C     GET LOGICAL UNITS FOR I/O DEVICES
C
      CALL RMPAR(IPAR)
      LIN=IPAR(1)
      LOUT=IPAR(2)
      NCNT=2
      ICNT=1
      IBLNK=IBLANK*256
```

```
C
C     RAD HEADING CARD (STYLE)
C
   10 READ(LIN,100) ICRD
      IF(ICRD(1).EQ.2HST) GOTO 1000
      WRITE(LOUT,200)
  200 FORMAT("    HEADING CARD MISSING ")
      GOTO 10
C
C     READ SIZE AREAS
C
   20 READ(LIN,100) (ICRD(I),I=1,10)
      DO 25 I=1,10
      IBUF(NCNT)=ICRD(I)
   25 NCNT=NCNT+1
      IF(ICRD(1).EQ.2H/E) GOTO 26
      IF(NCNT.NE.122) GO TO 20
   26 CALL EXEC(15,66,IBUF,128,MBF,NSCTR)
      NSCTR=NSCTR+1
      ICNT=ICNT+1
      NCNT=2
      DO 27 I=1,28
   27 IBUF(I)=0
      IF(ICRD(1).EQ.2H/E) GOTO 45
      GO TO 20
C
C    READ THE MARKERS
C
   45 NCNT=1
   50 READ(LIN,100) (ICRD(I),I=1,36)
      DO 90 I=1,36
      IF(ICRD(I).EQ.2H  ) GO TO 95
   90 CONTINUE
      I=36
   95 K=I-1
  150 FORMAT(" ICRD ",36A2)
      IF(ICRD(1).EQ.2H/E) GOTO 1020
  100 FORMAT(36A2)
      IF(ICRD(K).EQ.2H, .OR.ICRD(K).EQ.2H/ .
     1OR.ICRD(K).EQ.2H* ) ICRD(K)=IBLNK+ICRD(K)/256
      KXS=0
      IF(NCNT+K.LE.128) GOTO 110
      KXS=NCNT+K-128
      K=128-NCNT
  110 DO 300 I=1,K
      NCNT=NCNT+1
  300 IBUF(NCNT)=ICRD(I)
      IF(NCNT.LT.128) GO TO 50
      CALL EXEC(15,66,IBUF,128,MBF,NSCTR)
      NSCTR=NSCTR+1
      ICNT=ICNT+1
```

```
      IF(KXS.EQ.0) GO TO 390
      KK=K+KXS
      K2=K+1
      DO 380 I=K2,KK
  380 IBUF(I-K+1)=ICRD(I)
  390 KXS=KXS+1
      NCNT=KXS
      II=KXS+1
      DO 400 I=II,128
  400 IBUF(I)=0
      GOTO 50
C
 1000 CALL DIREC(2004B,ICRD(2),INTNO,NSCTR,ISTAT)
      CALL FLPAK(4)
      CALL DIREC(404B,ICRD(2),INTNO,NSCTR,ISTAT)
      IF(ISTAT.NE.0) GOTO 1010
      IFSTR=NSCTR
      GOTO 20
C
C     ERROR MESSAGE
C
 1010 WRITE(LOUT,250) ISTAT
 250  FORMAT("   BAD STATUS MBANK =",I3)
      GOTO 50
C
C    TERMINATOR  -  CLEAN UP
C
 1020 IF(NCNT.EQ.2) GOTO 1030
      CALL EXEC(15,66,IBUF,128,MBF,NSCTR)
C
C     UPDATE FIRST SECTOR
C
 1025 CALL EXEC(14,66,IBUF,128,MBF,IFSTR)
      IBUF(1)=ICNT
      CALL EXEC(15,66,IBUF,128,MBF,IFSTR)
C
C     LOAD MONITOR
C
      CALL EXEC(10,IP)
 1030 ICNT=ICNT-1
      GOTO 1025
      END
      END$
::
:JF,ST
:PR,FTN4,2,1,99
:LU,7,0
:PR,LOADR,2,7
RLIB
%MLIB,%CLIB,RDIRC,RFLPA,/E
:ST,P
:TY
**** LIST END ****
```

:LI,S,1,FLPG

```
:JO
:PU,ST,LPG
:CL
:ST,S,ST,10
FTN4
      PROGRAM LPG
C
C     ***************************************************
C     *                                                 *
C     *                 PROGRAM LPG                     *
C     *                                                 *
C     ***************************************************
C
C     LPG IS RESPONSIBLE FOR GENERATING THE LP MODEL AND
C     STORING THE RESULTS IN DISC FILES
C
C
C     ***********************************************
C     *                                             *
C     *          DISC FILE DOCUMENTATION            *
C     *                                             *
C     ***********************************************
C
C     THE READING AND WRITTING ON THE DISC PROCEDURES AND THE
C     SPACE ORGANIZATION IN THE DISC FILES ARE DEPENTED ON THE
C     ARRAY SIZES. HENCE ANY CHANGE ON THE CURRENT MEMORY SET-
C     UP MUST BE FOLLOWED BY THE PROPER ADJUSTMENT OF THESE
C     PROCEDURES AND FILES
C
C     A:   GENINF
C
C          SPECIFICATIONS :       TYPE OF DATA : BINARY
C                                 LENGTH       : 13 SECTORS
C                                                (CURRENT SET-UP)
C                                 OPERATION    : OVER-WRITTING
C                                                NO DIRECTORY
C
C          DISC FILE GENINF HAS THE FOLLOWING STRUCTURE
C
C     1. IDISC(4+4N3) :  EQUIVALENCED TO
C        DME          :  DUMMY MARKER EFFICIENCY
C        IS           :  NUMBER OF SIZES IN REQ. FILE
C        IPG          :  NUMBER OF FABRIC WIDTHS
C        PGW(N3)      :  FABRIC WIDTHS
C        PGQ(N3)      :  FABRIC QUANTITIES IN YDS ON STOCK
C                          ( FOR EVERY WIDTH )
C                        RESERVED SPACE : 1 SECTOR
```

```
C      2. JW(N2)          :  IDENTIFIES WIDTH FOR REAL MARKERS
C                              ( JW(J)=I, J-TH MARKER MADE FOR
C                                  I-TH WIDTH )
C                           RESERVED SPACE : 1 SECTOR
C      3. AR(N1)          :  PATTERN AREA ( FOR EVERY SIZE )
C                           RESERVED SPACE : 2 SECTORS
C      4. MKID1(N2),MKID2(N2),MKID3(N2),MKID4(N2),
C         MKID5(N2)      :  MARKER ID . TWO CHARACTERS PER WORD
C                           RESERVED SPACE : 5 SECTORS
C
C      ************************************************************
C
C      C.   ITMP1
C
C           SPECIFICATIONS :          TYPE OF DATA : BINARY
C                                     LENGTH       : 2+(N2/2) SCTS
C                                     OPERATION    : OVER-WRITTING
C                                                    NO DIRECTORY
C
C          DISC FILE ITMP1 HAS THE FOLLOWING STRUCTURE
C
C      1. RR(N1+N3)        : QUANTITIES OF GARMENTS REQUIRED
C                            ( RIGHT HAND SIDE OF THE MODEL )
C                            EQUIVALENCED TO RR1,RR2
C                           RESERVED SPACE : 2 SECTORS
C
C      2. IAWRK(N2)        : WORKING AREA FOR STORING ROW-WISE THE
C                            MATRIX MA ( IA FOR LPG ), WHERE ,
C          MA(N1,N2)  :  COEFFICIENTS OF THE PROBLEM
C                           RESERVED SPACE : (N2/2) SECTORS
C
C      *************************************************************
C
C      D.   ITMP2
C
C          SPECIFICATIONS :          TYPE OF DATA : BINARY
C                                    OPERATION    : OVER-WRITTING
C                                                   NO DIRECTORY
C
C          DISC FILE ITMP2 HAS THE FOLLOWING STRUCTURE
C
C      1. LPT(3+N5)       : BUFFER AREA WITH THE FOLOWING EQUIVALENCES
C           LPT(1)=NR   : NUMBER OF ROWS
C           LPT(2)=NV(MRKR IN LPG) : NUMBER OF VARIABLES (MARKERS)
C           LPT(3)=NRM :  NUMBER OF REAL MARKERS
C                           THE PART OF LPT BELOW THE THIRD WORD IS
C                           EQUIVALENCED TO LP WHERE,
C           LP(N5)       : NUMBER OF NONZERO ELEMENTS IN EACH COLUMN
C                            OF THE MATRIX MA (IA FOR LPG )
C
C      2. KC(N1+N3)        : IDENTIFIES THE TYPE OF CONSTRAINT
```

```
C
C      3. CC(N4)          : OBJECTIVE FUNCTION COEFFICIENTS
C                           (FABRIC AREA )
C
C      4. RR(N1+N3)        : REQUIRED GARMENT QUANTITIES + FABRIC
C                           STOCK QUANTINTIES
C
C      5. P(N8)           : NON-ZERO ELEMENTS OF THE MATRIX  MA
C
C      6. IP(N8)          : ROW POINTERS FOR EACH ELEMENT OF P
C
C         LP,P,IP ARE RESPECTIVLY LQ,Q,IQ IN LP8
C
C      **********************************************************
C
C
       INTEGER GENINF
       DIMENSION ISTYLE(5),ISZ1(60),ISZ2(60),ISZ3(60),RR(128),
      &IBUF(12),INAME(5),PGW(10),PGQ(10),MKID1(100),
      &MKID2(100),MKID3(100),MKID4(100),MKID5(100),ICRD(10),
      &IA(60,100),AR(100),CC(128),LP(125),RR1(64),RR2(64),ITMP2(3),
      &LPT(128),CC1(64),CC2(64),BFR(64),IBFR(128),KC(128),P(500),
      &IP(500),IWRK1(128),IWRK2(128),ICTID(10),IRFILE(3),JW(128)
       DIMENSION MBANK(3),IAWRK(120),ITMP1(3),IDISC(44),AR2(36),
      &GENINF(3),MONTR(3),EF(128)
C
       COMMON IDUM(2),IWORK(254),IDUMM(2),IBUF
       EQUIVALENCE (IWORK(1),IWRK1(2)),(IWORK(128),IWRK2(1))
       EQUIVALENCE (CC(1),CC1),(CC(128),CC2(64))
       EQUIVALENCE (LPT(4),LP)
       EQUIVALENCE (RR(1),RR1),(RR(128),RR2(64))
       EQUIVALENCE (BFR(1),IBFR)
       EQUIVALENCE (IDISC(1),EFF),(IDISC(3),ISCNT),(IDISC(4),IPG),
      &                 (IDISC(5),PGW),(IDISC(25),PGQ)
       EQUIVALENCE (AR(65),AR2)
C
       DATA AR,KC,ISZ1,ISZ2,ISZ3/100*0.,128*0,180*2H  /
       DATA ISLSH,IBLANK,IY,IYY,NULL/2H/ ,2H  ,2HY ,2HY ,2H  /
       DATA ITMP2/2HIT,2HMP,2H2 /
       DATA MBANK/2HMB,2HAN,2HK /
       DATA IRFILE,JW/2HRF,2HIL,2HE ,128*0/
       DATA ITMP1,GENINF/2HIT,2HMP,2H1 ,2HGE,2HNI,2HNF/
       DATA LOUT,LIN/1,1/
       DATA MONTR/2HCO,2HR ,2H  /
C
       ISCNT=0
       IPG=0
       MRKR=0
       NULL=(NULL*256)*256
       IYY=(IYY/256)*256
```

```
C
C     GET THE CUT ID
C
      READ(LIN,5) (ICRD(I),I=1,10)
    5 FORMAT(30A2)
      IF(ICRD(1).NE.2HCT) GO TO 90
      DO 80 I=1,5
   80 ICTID(I)=ICRD(1+I)
      GOTO 93
   90 WRITE(LOUT,3)
    3 FORMAT("  CUT ID CARD MISSING")
C
C    RETURN CONTROL TO MONITOR
      GOTO 640
C
   93 CALL DIREC(1005B,ICTID,INTNO,ISECT,NSTAT)
      IF(NSTAT.EQ.0) GOTO 95
      WRITE(LOUT,2)
    2 FORMAT("  CUT ID NOT IN DIRECTORY")
C
C    RETURN CONTROL TO MONITOR
      GOTO 640
C
C     *******  REQ. FILE ANALYSIS  *******
C
C
   95 CALL EXEC(14,66,IWORK,128,IRFILE,ISECT)
      IF(IWORK(2).EQ.2HST) GO TO 100
      WRITE(LOUT,8) ICTID
    8 FORMAT("  STYLE CARD MISSING FROM REQ. FILE : ",5A2)
C
C    RETURN CONTROL TO MONITOR
      GOTO 640
C
  100 DO 101 I=1,5
  101 ISTYLE(I)=IWORK(2+I)
      INDEX=0
      MTERM=1
  120 IF(IABS(INDEX).LT.112) GOTO 125
      ISECT=ISECT+1
      MTERM=1
      CALL EXEC(14,66,IWORK,128,IRFILE,ISECT)
  125 ISTRT=MTERM
      MTERM=MTERM+10
      INDEX=0
      MTWO=ISTRT+2
      IF(IWORK(ISTRT+1).EQ.2HSZ) GOTO 130
      IF(IWORK(ISTRT+1).EQ.2HPG) GOTO 180
      IF(IWORK(ISTRT+1).EQ.2HST) GOTO 125
      IF(IWORK(ISTRT+1).EQ.2H/E) GOTO 209
```

```
      IREC=ISCNT+IPG+2
      WRITE(LOUT,20) IWORK(ISTRT+1),ICTID,IREC
   20 FORMAT(" ILLEGAL SYNTAX : ",A2," IN REQ. FILE : ",5A2,
     &       /,21X,"LINE # ",I3)
C
C     RETURN CONTROL TO MONITOR
      GOTO 640
C
  130 ISCNT=ISCNT+1
      IF(IWORK(MTWO).EQ.IBLANK) GOTO 135
  132 WRITE(LOUT,25) ICTID
   25 FORMAT(" ILLEGAL SYNTAX IN REQ. FILE : ",5A2)
C
C     RETURN CONTROL TO MONITOR
      GOTO 640
C
  135 CALL LKTCH(2H, ,INDEX,MTERM,MTWO)
      IF(INDEX) 137,132,140
C
C     LEFT JUSTIFIED COMMA
C
  137 IF(IABS(INDEX).EQ.MTWO+1) GOTO 132
      IF(IABS(INDEX).GT.MTWO+1) ISZ1(ISCNT)=IWORK(MTWO+1)
      IF(IABS(INDEX).GT.MTWO+2) ISZ2(ISCNT)=IWORK(MTWO+2)
      IF(IABS(INDEX).GT.MTWO+3) ISZ3(ISCNT)=IWORK(MTWO+3)
      GOTO 160
C
C     RIGHT JUSTIFIED COMMA
C
  140 IBLNK=IBLANK
      IWORK(INDEX)=(IWORK(INDEX)/256)*256+IBLNK/256
      IF(INDEX.GE.MTWO+1) ISZ1(ISCNT)=IWORK(MTWO+1)
      IF(INDEX.GE.MTWO+2) ISZ2(ISCNT)=IWORK(MTWO+2)
      IF(INDEX.GE.MTWO+3) ISZ3(ISCNT)=IWORK(MTWO+3)
C
  160 IOLD=INDEX
      INDEX=-MTERM
      CALL LKTCH(2H, ,INDEX,MTERM,IABS(IOLD))
      IF((INDEX+IOLD).EQ.-1) GO TO 132
      CALL VAL(IOLD,INDEX,IERR,VALUE)
      IF(IERR.EQ.1) GO TO 132
      RR(ISCNT)=VALUE
      IF(INDEX.EQ.-MTERM) GOTO 120
      INDY=0
      CALL LKTCH(2HY ,INDY,MTERM,IABS(INDEX)-1)
      IF(INDY) 170,120,170
  170 KC(ISCNT)=1
      GOTO 120
```

```
C
  180 CALL LKTCH(2H, ,INDEX,MTERM,MTWO)
      I1=INDEX
      CALL LKTCH(2H  ,INDEX,MTERM,IABS(INDEX))
      IF(INDEX.EQ.I1.OR.(INDEX+I1).EQ.-1) GO TO 132
      CALL VAL(MTWO,I1,IERR,VALUE)
      IF(IERR.EQ.1) GO TO 132
      IPGG=IPG+1
      PGW(IPGG)=VALUE
      CALL VAL(I1,INDEX,IERR,VALUE)
      IF(IERR.EQ.1) GO TO 132
  185 PGQ(IPGG)=VALUE
      IPG=IPGG
      GOTO 120
C
C     *******  MARKER BANK ANALYSIS  *******
C
C
C     GET DUMMY MARKER EFFICIENCY
C
  209 WRITE(LOUT,27)
   27 FORMAT("  DUMMY MARKER EFFICIENCY :")
      READ(LIN,*) EFF
C
      CALL DIREC(1004B,ISTYLE,INTNO,ISECT,NSTAT)
      IF(NSTAT.EQ.0) GOTO 210
      WRITE(LOUT,15) ISTYLE
   15 FORMAT(" STYLE : ",5A2," NOT IN DIRECTORY")
C
C     RETURN CONTROL TO MONITOR
C
      GOTO 640
C
C     ***  PATTERN AREA SECTION  ***
C
  210 MTERM=0
      CALL EXEC(14,66,IWRK1,128,MBANK,ISECT)
  215 IF(IABS(INDEX).LT.111) GO TO 216
      ISECT=ISECT+1
      MTERM=0
      CALL EXEC(14,66,IWRK1,128,MBANK,ISECT)
  216 ISTRT=MTERM
      MTERM=MTERM+10
      IF(IWORK(ISTRT+1).EQ.2H/E) GOTO 240
      INDEX=0
      CALL LKTCH(2H, ,INDEX,MTERM,ISTRT)
      IF(INDEX.EQ.0) GO TO 230
      CALL PIKID(INDEX,6,ISTRT)
      IBT1=IBUF(1)+IBUF(2)
      IBT2=IBUF(3)+IBUF(4)
      IBT3=IBUF(5)+IBUF(6)
```

```
      DO 220 I=1,ISCNT
      IF(ISZ1(I).EQ.IBT1.AND.ISZ2(I).EQ.IBT2.AND.
     1   ISZ3(I).EQ.IBT3) GO TO 225
  220 CONTINUE
      GOTO 215
  225 IOLD=INDEX
      CALL LKTCH(2H  ,INDEX,MTERM,IABS(IOLD))
      CALL VAL(IOLD,INDEX,IERR,VALUE)
      IF(IERR.EQ.1) GO TO 230
      AR(I)=VALUE
      GOTO 215
  230 WRITE(1,30) ISTYLE
   30 FORMAT(" ILLEGAL SYNTAX IN MARKER BANK : ",5A2)
      GOTO 215
  240 DO 242 I=1,ISCNT
      IF(AR(I).NE.0.) GOTO 242
      WRITE(LOUT,35) ISZ1(I),ISZ2(I),ISZ3(I)
   35 FORMAT("  ENTER THE AREA OF THE SIZE : ",3A2)
      READ(LIN,*) AR(I)
  242 CONTINUE
C
C     ***  MARKER SECTION  ***
C
      ISECT=ISECT+1
      CALL EXEC(14,66,IWRK1,128,MBANK,ISECT)
      ISECT=ISECT+1
      CALL EXEC(14,66,IWRK2,128,MBANK,ISECT)
      DO 245 I=2,128
  245 IWORK(126+I)=IWRK2(I)
      IEND2=0
      IND5=254
      CALL LKTCH(2H@ ,IND5,254,1)
      IND4=0
  305 INDT=IND4
      CALL LKTCH(2H/ ,IND4,IABS(IND5),IABS(INDT))
      IF(IABS(IND5)-IABS(IND4).LE.3) IEND2=1
      IEND1=0
      IF(IND4.NE.INDT) GO TO 316
      IEND2=1
  314 IND4=IND5
  316 CALL LKTCH(2H, ,INDEX,254,IABS(INDT))
      MRKR=MRKR+1
      AREA=0.
```

```
C
C      START PARSING THIS MARKER
C
       CALL PIKID(INDEX,10,INDT)
       MKID1(MRKR)=IBUF(1)+IBUF(2)
       MKID2(MRKR)=IBUF(3)+IBUF(4)
       MKID3(MRKR)=IBUF(5)+IBUF(6)
       MKID4(MRKR)=IBUF(7)+IBUF(8)
       MKID5(MRKR)=IBUF(9)+IBUF(10)
  310  IOLD=INDEX
C      GET THE WIDTH OF THE MARKER
       CALL LKTCH(2H, ,INDEX,254,IABS(INDEX))
       CALL VAL(IOLD,INDEX,IERR,VALUE)
       WIDTH=VALUE
C      GET THE LENGTH OF THE MARKER
       II=IABS(INDEX)
       IOLD=INDEX
       CALL LKTCH(2H, ,INDEX,254,II)
       CALL VAL(IOLD,INDEX,IERR,VALUE)
       RLONG=VALUE
C
C      GET THE NEXT SIZE OF THIS MARKER
C     IF OVERFLOW IN  IWORK SHIFT LOWER SECTOR
C     BACK 128 WORDS
C
       IF(IABS(INDEX).LE.128.OR.IND5.NE.254) GOTO 318
       J2=IABS(INDEX)
       J3=254-J2
       DO 315 J=1,J3
  315  IWORK(128-J)=IWORK(255-J)
       IWORK(J2-127)=IWORK(J2)
       INDEX=(J2-127)*INDEX/J2
       ISECT=ISECT+1
       CALL EXEC(14,66,IWRK2,128,MBANK,ISECT)
       DO 317 I=2,128
  317  IWORK(126+I)=IWRK2(I)
       IND4=(IABS(IND4)-127)*IND4/IABS(IND4)
       CALL LKTCH(2H@ ,IND5,254,128)
  318  IOLD=INDEX
       INDEX=0
       CALL LKTCH(2H, ,INDEX,IABS(IND4),IABS(IOLD))
       IF(INDEX.NE.0) GO TO 319
       IEND1=1
       INDEX=IND4
  319  IND2=IOLD
       CALL LKTCH(2H* ,IND2,IABS(INDEX),IABS(IOLD))
       IND3=IND2
```

```
C
      CALL PIKID(INDEX,10,IND3)
C
      IBT1=IBUF(1)+IBUF(2)
      IBT2=IBUF(3)+IBUF(4)
      IBT3=IBUF(5)+IBUF(6)
      DO 340 K=1,ISCNT
   97 FORMAT(2(2X,@6,2X,A2))
      IF(ISZ1(K).EQ.IBT1.AND.ISZ2(K).EQ.IBT2.AND.
     *   ISZ3(K).EQ.IBT3) GO TO 350
  340 CONTINUE
C     NOT FOUND - GO TO CLEAR AREAS
      GOTO 400
C     GET THE NUMERIC ENTRY
  350 VALUE=1.
      IF(IND2.EQ.IOLD) GO TO 360
      CALL VAL(IOLD,IND2,IERR,VALUE)
  360 IA(K,MRKR)=INT(VALUE)
      AREA=AREA+AR(K)*VALUE
C
C     GET NEXT SIZE
C
      IF(IEND1.NE.1) GO TO 318
      CC(MRKR)=RLONG*WIDTH/36
      IF(IPG.EQ.0) GOTO 367
      DO 366 I=1,IPG
      IF(WIDTH.EQ.PGW(I)) JW(MRKR)=I
  366 CONTINUE
      EF(MRKR)=AREA/CC(MRKR)*100.
      IF(EF(MRKR).LT.EFF) GOTO 400
  367 IF(IEND2.EQ.1) GO TO 420
      GOTO 305
C
C     CLEAR WORKING AREAS
C
  400 DO 410 K=1,ISCNT
  410 IA(K,MRKR)=0
      MKID1(MRKR)=IBLANK
      MKID2(MRKR)=IBLANK
      MKID3(MRKR)=IBLANK
      MKID4(MRKR)=IBLANK
      MKID5(MRKR)=IBLANK
      MRKR=MRKR-1
C
C    GET NEXT MARKER
C
      GOTO 367
```

```
C
C        CREATE THE DISC FILE ITMP1
C        IF THE SIZE OF (RR1+RR2=RR) IS GREATER THAN 128 THE
C        FOLLOWING LINES MUST BE EXPANDED
C
  420   CALL EXEC(15,66,RR1,128,ITMP1,0)
        IF(ISCNT.GT.64) CALL EXEC(15,66,RR2,128,ITMP1,1)
        ISECT=1
C        THE FOLLOWING SECTION WORKS ONLY FOR A 60X100 MATRIX
        DO 425 K=1,99,2
        ISECT=ISECT+1
        DO 422 I=1,60
        IAWRK(I)=IA(I,K)
  422   IAWRK(I+60)=IA(I,K+1)
        CALL EXEC(15,66,IAWRK,120,ITMP1,ISECT)
  425   CONTINUE
        ISECT=ISECT+1
C
C     DUMMY MARKERS
C
        NRM=MRKR
        IF(EFF.EQ.0.) GO TO 460
        DO 450 I=1,ISCNT
        MRKR=MRKR+1
  430   CC(MRKR)=AR(I)/EFF
        MKID1(MRKR)=ISZ1(I)
        MKID2(MRKR)=ISZ2(I)
        MKID3(MRKR)=ISZ3(I)
  450   IA(I,MRKR)=1
C
C     ADJUST MATRIX FOR CUTDOWNS
C
  460   DO 470 I=2,ISCNT
        IF(KC(I).EQ.0) GO TO 470
        DO 465 J=1,MRKR
  465   IA(I,J)=IA(I,J)+IA(I-1,J)
        RR(I)=RR(I)+RR(I-1)
  470   KC(I-1)=KC(I)
        KC(ISCNT)=0
        CALL EXEC(15,66,KC,128,ITMP1,ISECT)
C
C       ADD FABRIC STOCK CONSTRAINTS
C
        NR=ISCNT
        IF(IPG.EQ.0) GOTO 474
        DO 473 I=1,IPG
        NR=NR+1
        RR(NR)=PGQ(I)
        AR(NR)=0.
  473   KC(NR)=1
  474   K=0
```

```
C
C      NONZERO ELEMENTS
C
       DO 480 J=1,MRKR
       DO 475 I=1,ISCNT
       IF(IA(I,J).LE.0) GO TO 475
       K=K+1
       IP(K)=I
       P(K)=FLOAT(IA(I,J))
 475   CONTINUE
       IF(JW(J).EQ.0) GOTO 480
       K=K+1
       IP(K)=ISCNT+JW(J)
       P(K)=36.*CC(J)/PGW(JW(J))
 480   LP(J)=K
C
C       THE FOLLOWING TWO SECTIONS (ITMP2,GENINF) MUST BE
C       ADJUSTED IF THE ARRAY SIZES WERE MODIFIED
C
C    CREATE THE DISC FILE ITMP2
C
       LPT(1)=NR
       LPT(2)=MRKR
       LPT(3)=NRM
       LW=MRKR+3
C
       CALL EXEC(15,66,LPT,LW,ITMP2,0)
       CALL EXEC(15,66,KC,NR,ITMP2,1)
       CALL EXEC(15,66,CC1,128,ITMP2,2)
       IF(MRKR.GT.64) CALL EXEC(15,66,CC2,128,ITMP2,3)
       CALL EXEC(15,66,RR1,128,ITMP2,4)
       IF(NR.GT.64) CALL EXEC(15,66,RR2,128,ITMP2,5)
C
       IB1=INT(2.*FLOAT(LP(MRKR))/128.)
       IB2=INT(FLOAT(LP(MRKR))/128.)
       ISECT=5
       JJ=0
       IF(IB1.EQ.0) GOTO 540
       DO 530 I=1,IB1
       ISECT=ISECT+1
       DO 520 J=1,64
   520 BFR(J)=P(JJ+J)
       CALL EXEC(15,66,BFR,128,ITMP2,ISECT)
       JJ=JJ+64
   530 CONTINUE
   540 IB1=LP(MRKR)-(128*INT(FLOAT(LP(MRKR))/128.))
       IB3=IB1
       IF(IB1.EQ.LP(MRKR)) IB1=LP(MRKR)-64
       IF(IB1) 550,570,550
   550 ISECT=ISECT+1
```

```
      DO 560 J=1,IB1
  560 BFR(J)=P(JJ+J)
      IB1=IB1+IB1
      CALL EXEC(15,66,BFR,IB1,ITMP2,ISECT)
C
  570 JJ=0
      IF(IB2.EQ.0) GOTO 610
      DO 600 I=1,IB2
      ISECT=ISECT+1
      DO 590 J=1,128
  590 IBFR(J)=IP(JJ+J)
      CALL EXEC(15,66,IBFR,128,ITMP2,ISECT)
      JJ=JJ+128
  600 CONTINUE
  610 IF(IB3.EQ.0) STOP
      ISECT=ISECT+1
      DO 630 J=1,IB3
  630 IBFR(J)=IP(JJ+J)
      CALL EXEC(15,66,IBFR,IB3,ITMP2,ISECT)
C
C    CREATE THE DISC FILE GENINF
C
      CALL EXEC(15,66,IDISC,44,GENINF,0)
      CALL EXEC(15,66,JW,128,GENINF,1)
      CALL EXEC(15,66,AR,128,GENINF,2)
      IF(ISCNT.GT.64) CALL EXEC(15,66,AR2,72,GENINF,3)
      CALL EXEC(15,66,ISZ1,60,GENINF,4)
      CALL EXEC(15,66,ISZ2,60,GENINF,5)
      CALL EXEC(15,66,ISZ3,60,GENINF,6)
      CALL EXEC(15,66,MKID1,100,GENINF,7)
      CALL EXEC(15,66,MKID2,100,GENINF,8)
      CALL EXEC(15,66,MKID3,100,GENINF,9)
      CALL EXEC(15,66,MKID4,100,GENINF,10)
      CALL EXEC(15,66,MKID5,100,GENINF,11)
      CALL EXEC(15,66,EF,256,GENINF,12)
C
C
C
C     RETURN CONTROL TO MONITOR
C
 640  CALL EXEC(10,MONTR)
      END
```

```
C
C     *****************************************************
C
      SUBROUTINE LKTCH(ICHAR,INDEX,IF,IB)
C
C     ****************************************
C     *          SUBROUTINE LKTCH             *
C     ****************************************
C
C     LKTCH LOCATES THE FIRST OCCURENCE OF A SPECIFIED CHAR. IN
C     A PACKED STRING OF CHARACTERS (TWO CHARACTERS PER WORD).
C     ONLY THOSE SUBSTRINGS ARE SEARCHED WHERE THE SPECIFIED CHA
C     MAY APPEAR. A NEGATIVE (POSITIVE) POINTER INDICATES A LEFT
C     (RIGHT) JUSTIFIED CHARACTER.
C
C     GLOSSARY :
C              1.  ICHAR :  CHARACTER TO BE LOCATED
C              2.  INDEX :  (+,-) POINTER FOR THE POSITION OF
C                           ICHAR IN THE SUBSTRING
C              3.  IB    :  LOWER BOUNDARY OF THE SUBSTRING
C              4.  IF    :  UPPER BOUNDARY OF THE SUBSTRING
C
C     *******************************************************
C
      COMMON IDUM(2),IWORK(254),IDUMM(2),IBUF(12)
      ICHAR=(ICHAR/256)*256
      IBB=IB+1
      DO 10 I=IBB,IF
      IW=IWORK(I)
      IF((IW/256)*256.NE.ICHAR) GO TO 5
      INDEX=-I
   20 FORMAT(" INDEX =",I4)
      RETURN
    5 IW=IWORK(I)
      IF(IW*256.NE.ICHAR) GO TO 10
      INDEX=I
      RETURN
   10 CONTINUE
      RETURN
      END
C
```

```
C
      SUBROUTINE PIKID(IDX,LIMIT,ISTRT)
C
C     ****************************************
C     *           SUBRUOTINE PIKID           *
C     ****************************************
C
C     PIKID  OPERATES ON A SPECIFIED PACKED STRING OF CHARACTERS.
C     THE BASIC FUNCTION OF THE SUBROUTINE IS TO UNPACK THE STRIN
C     AND STORE THE CHARACTERS IN A BUFFER (CHAR. BY CHAR.).
C     THE ODD (EVEN) NUMBER OF CHAR. ARE STORED AS LEFT(RIGHT)
C     JUSTIFIED. ONLY THE SPECIFIED MAX NUMBER OF CHARACTERS
C     IS UNPACKED AND THE REST IS IGNORED.
C     PIKID IS USED BY THE CALLING PROGRAM IN THE PROCESS OF
C     NAME IDENTIFICATION.
C
C     GLOSSARY :
C                  1.  IDX    :  UPPER BOUNDARY OF THE STRING
C                  2.  LIMIT :  MAX NUMBER OF CHARACTERS IN A NAME
C                  3.  ISTRT :  LOWER BOUNDARY OF THE STRING
C                  4.  IBUF   :  BUFFER AREA WHERE THE UNPACKED
C                                CHARACTERS ARE STORED
C
C     *****************************************************************
C
      COMMON IDUM(2),IWORK(254),IDUMM(2),IBUF(12)
      IBLANK=2H
      NULL=1
      DO 10 I=1,10
   10 IBUF(I)=IBLANK
      IK=IABS(IDX)
      IF(IDX.GT.0) IWORK(IK)=(IWORK(IK)/256)*256
      IF(IDX.LT.0) IK=IK-1
      JK=IABS(ISTRT)
      JP=0
      IF(ISTRT.GE.0) GO TO 20
      JP=JP+1
      IBUF(JP)=IWORK(JK)*256
   20 IF(JK.EQ.IK.OR.JP.EQ.LIMIT) GO TO 30
      JP=JP+1
      JK=JK+1
      IW1=IWORK(JK)
      IBUF(JP)=(IW1/256)*256
      IF(JP.EQ.LIMIT) GO TO 30
      JP=JP+1
      IBUF(JP)=IWORK(JK)*256
      GOTO 20
   30 IF(IBUF(JP).EQ.(NULL*256)*256) IBUF(JP)=2H
      DO 40 I=1,9,2
   40 IBUF(I)=(IBUF(I)/256)*256
      DO 50 I=2,10,2
   50 IBUF(I)=IBUF(I)/256
      RETURN
      END
```

```
C
C
      SUBROUTINE VAL(IT,INDEX,IERR,VALUE)
C
C     ****************************************
C     *           SUBROUTINE VAL             *
C     ****************************************
C
C     VAL CONVERTS A PACKED ALPHA STRING INTO A NUMERIC.
C     AT FIRST THE STRING IS UNPACKED AND EACH CHARACTER IS
C     STORED (LEFT JUSTIFIED) INTO A WORKING AREA (ONE CHARACTER
C     PER WORD). THEN EACH CHARACTER IS CONVERTED TO THE CORRES-
C     PONDING NUMERIC AND THE VALUE OF THE STRING IS EVALUATED
C
C     GLOSSARY :
C              1. IT    :  LOWER  BOUNDARY OF THE  STRING
C              2. INDEX :  UPPER BOUNDARY OF THE STRING
C              3. IERR  :  ERROR FLAG ( 0,1 ). IF IERR=1
C                          ILLEGAL CHARACTER IN NUMERIC FIELD
C              4. VALUE :  VALUE OF THE NUMERIC
C              5. LN    :  CONTAINS THE ALPHA REPRESENTATION
C                          OF THE NUMBERS 0-9
C
C     ***********************************************************
C
      DIMENSION IW1(15),LN(10)
      COMMON IDUM(2),IWORK(254),IDUMM(2),IBUF(12)
      DATA LN/2H0 ,2H1 ,2H2 ,2H3 ,2H4 ,2H5 ,2H6 ,2H7 ,2H8 ,2H9 /
      DATA IBLANK,IPRD,NULL/2H  ,2H. ,2H  /
      DATA LOUT/1/
      NULL=(NULL*256)*256
      IBLANK=(IBLANK/256)*256
      IPRD=(IPRD/256)*256
      IOLD=IT
      DO 5 I=1,10
    5 LN(I)=(LN(I)/256)*256
C
C     UNPACK THE ALPHA STRING
C
      K=0
      IERR=0
      IF(IOLD.GT.0) GO TO 10
      IOLD=IABS(IOLD)
      IW1(1)=IWORK(IOLD)
      IW1(1)=IW1(1)*256
      K=1
   10 IOLD=IOLD+1
      IE=IABS(INDEX)-1
      IF(IE.LT.IOLD) GO TO 17
```

```
      DO 15 J=IOLD,IE
      K=K+1
      IW=IWORK(J)
      IW1(K)=(IW/256)*256
      K=K+1
      IW1(K)=IWORK(J)
      IW1(K)=IW1(K)*256
   15 CONTINUE
  17  IF(INDEX.LT.0) GO TO 20
      K=K+1
      IW1(K)=IWORK(IE+1)
      IW1(K)=(IW1(K)/256)*256
C
C     CONVERT THE UNPACKED ALPHA STRING TO NUMERIC
C
   20 VALUE=0.
      LDP=K
      DO 40 I=1,K
      DO 30 J=1,10
   90 FORMAT(" IW= ",@6,5X,A2)
      IF(IW1(I).EQ.LN(J)) GO TO 35
   30  CONTINUE
      IF(IW1(I).EQ.IBLANK.OR.IW1(I).EQ.NULL) GOTO 40
      IF(IW1(I).NE.IPRD) GO TO 33
      LDP=I
      GOTO 40
  33  WRITE(1,100)
   100 FORMAT(" FATAL ERROR : ILLEGAL CHARACTER IN NUM. FIELD")
      IERR=1
      RETURN
  35  VALUE=10.*VALUE+FLOAT(J-1)
   40  CONTINUE
      VALUE=VALUE*.1**(K-LDP)
      RETURN
      END
      END$
C
C
::
:JF,ST
:PR,FTN4,2,1,99
:LU,7,0
:PR,LOADR,2,7
RLIB
%MLIB,%CLIB,RDIRC,RFLPA,/E
:ST,P
:TY
**** LIST END ****
```

```
:LI,S,1,FLP8

:JO
:PU,ST,LP8
:CL
:ST,S,ST,10
FTN4
      PROGRAM LP8
C
C     **************************************************
C     *                                                *
C     *               PROGRAM LP8                      *
C     *                                                *
C     **************************************************
C
C
C     LP8 IS THE OPTIMIZATION PROGRAM. IT USES A LINEAR INTEGER
C     PROGRAMMING ALGORITHM FOR FINDING AN INTEGER SOLUTION.
C     THE SOLUTION IS STORED IN FILE OUTINF WHICH IS AN INPUT
C     FILE TO LPM2 ( PRODUCTION REPORT ).
C       R(2(N1+N3))     :  CONSTRAINT VALUE
C       KC(2(N1+N3))    :  TYPE OF CONSTRAINT
C       BIS             :  BASIC INTEGER SOLUTION
C       NPS             :  # OF ADDITIONAL CONSTRAINTS IN STACK
C       ISM             :  ORDER INDEX OF VAR WITH SMALLEST RE-
C                          MAINDER
C       KCS(N4)         :  KIND OF CONSTRAINT
C       CS(N4)          :  VALUE OF CONSTRAINT
C       ICS(N4)         :  INDEX OF CONSTRAINED VARIABLE--CARRIER
C                          OF BRANCHING INFORMATION
C       IPS(N4)         :  POINTER TO LOCATION IN CONSTRAINT STACK
C                          WHERE CONSTRAINT OF J-TH VAR IS STORED
C
C     ************************************************
C     *                                              *
C     *           DISC FILE DOCUMENTATION            *
C     *                                              *
C     ************************************************
C
C     THE READING AND WRITTING ON THE DISC PROCEDURES AND THE
C     SPACE ORGANIZATION IN THE DISC FILES ON THE DISC ARE
C     DEPENTED ON THE ARRAY SIZES. HENCE ANY CHANGE ON THE
C     CURRENT MEMORY SET-UP MUST BE FOLLOWED BY THE PROPER
C     ADJUSTMENT OF THESE PROCEDURES AND FILES.
C
```

```
C
C     A:   GENINF
C
C          SPECIFICATIONS :        TYPE OF DATA : BINARY
C                                  LENGTH       : 13 SECTORS
C                                                 (CURRENT SET-UP)
C                                  OPERATION    : OVER-WRITTING
C                                                 NO DIRECTORY
C
C          DISC FILE GENINF HAS THE FOLLOWING STRUCTURE
C
C     1. IDISC(4+4N3) :  EQUIVALENCED TO
C        DME          :  DUMMY MARKER EFFICIENCY
C        IS           :  NUMBER OF SIZES IN REQ. FILE
C        IPG          :  NUMBER OF FABRIC WIDTHS
C        PGW(N3)      :  FABRIC WIDTHS
C        PGQ(N3)      :  FABRIC QUANTITIES IN YDS ON STOCK
C                          ( FOR EVERY WIDTH )
C                        RESERVED SPACE : 1 SECTOR
C
C     2. JW(N2)       :  IDENTIFIES WIDTH FOR REAL MARKERS
C                          ( JW(J)=I, J-TH MARKER MADE FOR
C                             I-TH WIDTH )
C                        RESERVED SPACE : 1 SECTOR
C
C     3. AR(N1)          : PATTERN AREA ( FOR EVERY SIZE )
C                        RESERVED SPACE : 2 SECTORS
C
C     4. MKID1(N2),MKID2(N2),MKID3(N2),MKID4(N2),
C        MKID5(N2)    :  MARKER ID . TWO CHARACTERS PER WORD
C                        RESERVED SPACE : 5 SECTORS
C
C     ***********************************************************
C
C     B:   OUTINF
C
C          SPECIFICATIONS :        TYPE OF DATA : BINARY
C                                  LENGTH       : 8 SECTORS
C                                                 (CURRENT SET-UP)
C                                  OPERATION    : OVER-WRITTING
C                                                 NO DIRECTORY
C
```

```
C          DISC FILE OUTINF HAS THE FOLLOWING STRUCTURE
C
C     1. KBB(N5)      :  POINTER LINKING ACTUAL COLUMN INDECES
C                        IN LP SOLUTION WITH ORIGINAL INDECES OF
C                        UKNOWN VARIABLES. IT IS EQUIVALENCED TO
C                        KBB1,KBB2,KBB3, FROM BELOW , WHERE THE
C                        FIRST 3 WORDS OF KBB1 ARE EQUIVALENCED TO
C           NRB : NUMBER OF MARKERS ( INCLUDING DUMMY )
C           TAT : VALUE OF OBJECTIVE FUNCTION ( TOTAL FABRIC AREA)
C           INIT: NUMBER OF INTEGER ITERATION STEPS. PRINTED IN
C                 PRODUCTION REPORT AS "... REAL AND INT SOLUTIONS
C                       RESERVED SPACE : 3 SECTORS
C
C     2. AB(N5)        :  RESULTING VALUES OF UNKNOWN VARIABLES
C                         EQUIVALENCED TO : AB1,AB2,AB3,AB4,AB5
C                       RESERVED SPACE : 5 SECTORS
C
C     ***********************************************************
C
C     C. ITMP2
C
C          SPECIFICATIONS :         TYPE OF DATA : BINARY
C                                   LENGTH       : 3+N1+N4+2N5+2N8
C                                   OPERATION    : OVER-WRITTING
C                                                  NO DIRECTORY
C
C          DISC FILE ITMP2 HAS THE FOLLOWING STRUCTURE
C
C     1. LPT(3+N5)     : BUFFER AREA WITH THE FOLOWING EQUIVALENCES
C           LPT(1)=NR   : NUMBER OF ROWS
C           LPT(2)=NV(MRKR IN LPG) : NUMBER OF VARIABLES (MARKERS)
C           LPT(3)=NRM :  NUMBER OF REAL MARKERS
C                        THE PART OF LPT BELOW THE THIRD WORD IS
C                        EQUIVALENCED TO LP WHERE,
C           LP(N5)      : NUMBER OF NONZERO ELEMENTS IN EACH COLUMN
C                         OF THE MATRIX MA (IA FOR LPG )
C
C     2. KC(N1+N3)      : IDENTIFIES THE TYPE OF CONSTRAINT
C
C     3. CC(2N5-N4)     : OBJECTIVE FUNCTION COEFFICIENTS
C                         (FABRIC AREA)
C
C     4. R(N1+N3)       : REQUIRED GARMENT QUANTITIES + FABRIC
C                        STOCK QUANTINTIES
C
C     5. P(N8)         : NON-ZERO ELEMENTS OF THE MATRIX  MA
C
C     6. IP(N8)        : ROW POINTERS FOR EACH ELEMENT OF P
C
C          LP,P,IP ARE RESPECTIVLY LQ,Q,IQ IN LP8
C
C     ***********************************************************
```

```
C
      DOUBLE PRECISION RATIO,RATMIN
      INTEGER OUTINF
      DIMENSION C(3ØØ),C1(64),C2(64),LQT(3Ø3),Q(3ØØØ),IQ(3ØØØ),
     &R(14Ø),KC(14Ø),R1(64),R2(64),BFR(64),IBFR(128),COLIN(3ØØ),
     &LQ(3ØØ),L(3ØØ),ITMP2(3),OUTINF(3),KBB1(128),KBB2(128),
     &KBB3(48),AB1(64),AB2(64),AB3(64),AB4(64),AB5(44),
     &IB(3ØØ),KB(3ØØ),CS(16Ø),KCS(16Ø),ICS(16Ø),IPS(16Ø)
      DIMENSION KBB(3ØØ),AB(3ØØ),IRGEN(3)
      COMMON Q,IDUM(4),LQ,IDUM2(4),IQ
C
      EQUIVALENCE (C(1),C1),(C(65),C2)
      EQUIVALENCE (R(1),R1),(R(65),R2)
      EQUIVALENCE (BFR(1),IBFR)
      EQUIVALENCE (LQT(4),LQ)
      EQUIVALENCE (KBB1(1),RKBB),(KBB1(128),KBB(124)),
     &(KBB2(128),KBB(252)),(KBB3(48),KBB(3ØØ))
      EQUIVALENCE (AB(1),AB1),(AB(65),AB2),(AB(129),AB3),
     &             (AB(193),AB4),(AB(257),AB5)
C
      DATA IRGEN/2HLP,2HM2,2H  /
      DATA OUTINF/2HOU,2HTI,2HNF/
      DATA EPS,NPS,IPS/1.E-4,2Ø1*Ø/
      DATA INP,LOUT/1,1/
      DATA ITMP2/2HIT,2HMP,2H2 /
      INIT=Ø
      BEST=1.E+12
  19Ø WRITE(LOUT,9)
    9 FORMAT("   FIRST-BEST INTEGER SOLUTION")
      READ(INP,1Ø) NNN
   1Ø FORMAT(A2)
      IF(NNN.NE.2HFI.AND.NNN.NE.2HBE) GOTO 19Ø
  2ØØ INIT=INIT+1
      K=Ø
C
C    GET THE MODEL FROM THE DISC FILE  ITMP2
C
C     THE FOLLOWING SECTION MUST BE ADJUSTED IF THE ARRAY SIZES
C     INVOLVED WERE MODIFIED.
C
      CALL EXEC(14,66,LQT,128,ITMP2,Ø)
      NR=LQT(1)
      NV=LQT(2)
      NRM=LQT(3)
      CALL EXEC(14,66,KC,NR,ITMP2,1)
      CALL EXEC(14,66,C1,128,ITMP2,2)
      IF(NV.GT.64) CALL EXEC(14,66,C2,128,ITMP2,3)
      CALL EXEC(14,66,R1,128,ITMP2,4)
      IF(NR.GT.64) CALL EXEC(14,66,R2,128,ITMP2,5)
```

```
      IB1=INT(2.*FLOAT(LQ(NV))/128.)
      IB2=INT(FLOAT(LQ(NV))/128.)
      ISECT=5
      JJ=0
      IF(IB1.EQ.0) GOTO 54
      DO 53 I=1,IB1
      ISECT=ISECT+1
      CALL EXEC(14,66,BFR,128,ITMP2,ISECT)
      DO 52 J=1,64
   52 Q(JJ+J)=BFR(J)
      JJ=JJ+64
   53 CONTINUE
   54 IB1=LQ(NV)-(128*INT(FLOAT(LQ(NV))/128.))
      IB3=IB1
      IF(IB1.EQ.LQ(NV)) IB1=LQ(NV)-64
      IF(IB1.EQ.0) GOTO 57
      ISECT=ISECT+1
      IB1=IB1+IB1
      CALL EXEC(14,66,BFR,IB1,ITMP2,ISECT)
      IB1=IB1/2
      DO 56 J=1,IB1
   56 Q(JJ+J)=BFR(J)
   57 JJ=0
      IF(IB2.EQ.0) GOTO 61
      DO 60 I=1,IB2
      ISECT=ISECT+1
      CALL EXEC(14,66,IBFR,128,ITMP2,ISECT)
      DO 59 J=1,128
   59 IQ(JJ+J)=IBFR(J)
      JJ=JJ+128
   60 CONTINUE
   61 IF(IB3.EQ.0) GOTO 64
      ISECT=ISECT+1
      CALL EXEC(14,66,IBFR,IB3,ITMP2,ISECT)
      DO 63 J=1,IB3
   63 IQ(JJ+J)=IBFR(J)
   64 CONTINUE
      INDEX=3000-LQ(NV)
      DO 5 J=1,LQ(NV)
      Q(INDEX+J)=Q(J)
 5    IQ(INDEX+J)=IQ(J)
```

```
C
C     MODIFY ORIGINAL LP PROBLEM BY INTEGER
C     PROGRAMMING CONSTRAINTS AND ADDING SLACK AND
C     ARTIFICIAL VARIABLES
C
      NC=NV
      IZ=INDEX
      DO 202 J=1,NC
      IA=IZ+1
      IZ=INDEX+LQ(J)
      DO 201 I=IA,IZ
      K=K+1
      Q(K)=Q(I)
  201 IQ(K)=IQ(I)
      IF (IPS(J).EQ.0) GO TO 2011
C
C     ADD INEGER PROGRAMMING CONSTRAINTS
C
      NR=NR+1
      R(NR)=CS(IPS(J))
      KC(NR)=KCS(IPS(J))
      IPS(J)=0
      K=K+1
      Q(K)=1.
      IQ(K)=NR
 2011 LQ(J)=K
  202 C(J)=-C(J)
C
C     ADD SLACK AND ARTIFICIAL VARIABLES
C
      DO 204 I=1,NR
      IB(I)=0
      J=1
  203 NC=NC+1
      K=K+1
      Q(K)=FLOAT(J)
      IQ(K)=I
      LQ(NC)=K
      C(NC)=0.
      IF(KC(I).LE.0.AND.J.GT.0) C(NC)=-1.E3
      J=J*KC(I)
      IF(J.LT.0) GO TO 203
  204 CONTINUE
```

```
C
C       ADD THE RIGHT HAND SIDE VECTOR TO THE MATRIX
C
        NC1=NC+1
        C(NC1)=0.
        DO 2041 I=1,NR
        K=K+1
        Q(K)=R(I)
 2041   IQ(K)=I
        LQ(NC1)=K
C ------------- GET INITIAL BASIS ---------
        J=NC1
        DO 207 K=1,NR
  205   J=J-1
        IF(J.LE.0) GO TO 208
        NOZER=0
        NOONE=0
        DO 206 I=1,NR
        IF(A(I,J).GT.-EPS.AND.A(I,J).LT.EPS) NOZER=NOZER+1
        IF(A(I,J).LE.1.-EPS.OR.A(I,J).GE.1.+EPS) GO TO 206
        NOONE=NOONE+1
        NOHIT=I
  206   CONTINUE
        IF(NOONE.NE.1.OR.NOZER.NE.NR-1.OR.IB(NOHIT).NE.0) GO TO 205
        COLIN(NOHIT)=C(J)
        KB(NOHIT)=J
  207   IB(NOHIT)=J
  208   DO 209 I=1,NR
        IF(IB(I).NE.0) GO TO 209
        WRITE (1,120) I
        GO TO 224
  209   CONTINUE
C ----------- ELIMINATE BASIC VARIABLES FROM OBJECTIVE FCN
        DO 211 J=1,NC1
        DO 210 I=1,NR
  210   C(J)=C(J)-COLIN(I)*A(I,J)
  211   C(J)=-C(J)
C ----------- BEGIN MAIN ITERATION LOOP
        IT=0
  221   INCOL=0
        CMIN=-EPS
        DO 222 J=1,NC
        IF(C(J).GE.CMIN) GO TO 222
        CMIN=C(J)
        INCOL=J
  222   CONTINUE
```

```
C ----------- PICK ROW TO PIVOT ON
      IF(INCOL.EQ.Ø) GO TO 3ØØ
      INROW=Ø
      RATMIN=99999999.
      DO 223 I=1,NR
      IF(A(I,INCOL).LE.EPS) GO TO 223
      RATIO=A(I,NC1)/A(I,INCOL)
      IF(RATIO.GE.RATMIN) GO TO 223
      RATMIN=RATIO
      INROW=I
  223 CONTINUE
C ------------- UNBOUNDED SOLUTION
      IF(INROW.NE.Ø) GO TO 225
  224 WRITE (INP,119)
      STOP
C ------------- PIVOT
  225 J=KB(INROW)
      KB(INROW)=INCOL
      DO 226 I=1,NR
  226 COLIN(I)=A(I,INCOL)
      CSTIN=C(INCOL)
      COEF=A(INROW,INCOL)
      N=LQ(NC1)
C
C     UPDATE THE STRINGS Q AND IQ AND MOVE THEM
C     TO THE END OF THE ARRAYS
C
      NN=3ØØØ
      J=NC1+1
  231 IF(J.EQ.1) GO TO 24Ø
      LQ(J)=NN
      J=J-1
      M=Ø
      IF(J.GT.1) M=LQ(J-1)
      IF(A(INROW,J).NE.Ø.) GO TO 233
  232 Q(NN)=Q(N)
      IQ(NN)=IQ(N)
      N=N-1
      NN=NN-1
      IF(N.GT.M) GO TO 232
      GOTO 231
  233 CORR=A(INROW,J)/COEF
      C(J)=C(J)-CSTIN*CORR
      I=NR+1
  234 IF(I.EQ.1) GO TO 231
      I=I-1
      IF(I.NE.INROW) GO TO 235
      Q(NN)=CORR
      GOTO 239
  235 IF(COLIN(I).EQ.Ø.) GO TO 238
```

```
C
C     CHECK THE GAP BETWEEN THE ACTIVE LOCATIONS
C     OF THE Q AND IQ ARRAYS FOR NON-OVERLAPPING
C
      IF(I.EQ.IQ(N).AND.N.GT.M) GO TO 237
      IF(N.LT.NN) GO TO 236
      WRITE(INP,130) J
 130  FORMAT(" DYNAMIC STORAGE OVERLAP AT J=",I3)
      STOP
 236  N=N+1
      Q(N)=Ø.
 237  Q(NN)=Q(N)-COLIN(I)*CORR
      GOTO 239
 238  IF(I.NE.IQ(N).OR.N.LE.M) GO TO 234
      Q(NN)=Q(N)
 239  IQ(NN)=I
      N=N-1
      IF (ABS(Q(NN)).GT.EPS) NN=NN-1
      GO TO 234
C
C     BRING THE STRINGS Q AND IQ BACK TO THE BEGGINING
C     OF THE ARRAYS
C
 24Ø  K=Ø
      NN=NN+1
      DO 241 I=NN,3ØØØ
      K=K+1
      Q(K)=Q(I)
 241  IQ(K)=IQ(I)
      DO 242 J=1,NC1
 242  LQ(J)=LQ(J+1)-3ØØØ+K
      IT=IT+1
      GO TO 221
C
C     OPTIMUM REAL OR INEGER SOLUTION REACHED ;
C     CHECK THE INTEGRALITY AND FORMULATE AN
C     ADDITIONAL CONSTRAINT
C
 3ØØ  IF(ABS(C(NC1)).GT.1E+5) GOTO 4Ø6
      L(1)=1
      DO 3Ø2 I=2,NR
      J=I-1
 3Ø1  IF (KB(I).GT.KB(L(J))) GO TO 3Ø2
      L(J+1)=L(J)
      J=J-1
      IF (J.GT.Ø) GO TO 3Ø1
 3Ø2  L(J+1)=I
```

```
C
C      SELECT THE NEXT INTEGER CONSTRAINT
C
       SMALL=1.
       I=1
   401 REM=AMOD(A(L(I),NC1),1.)
   12  FORMAT(2(2X,I4),2(2X,E10.3))
       IF (REM.GE.SMALL.OR.REM.LE.EPS.OR.REM.GE.1.-EPS) GO TO 402
       SMALL=REM
       ISM=L(I)
   402 I=I+1
       IF(KB(L(I)).LE.NRM.AND.I.LE.NR) GOTO 401
       TEMP=-C(NC1)
       IF(SMALL.LT.FLOAT(1).AND.TEMP.LT.BEST) GO TO 410
       IF (SMALL.LT.1.)GO TO 406
       IF (-C(NC1).GE.BEST) GO TO 406
       BEST=-C(NC1)
       DO 405 I=1,NR
       KBB(I)=KB(L(I))
       AB(I)=A(L(I),NC1)
   405 CONTINUE
       FCNB=-C(NC1)
       I=2*(NR+1)
       ISECT=1+INT(FLOAT(NR)/128.)
       ITB=IT
       NRB=NR
       IF(NNN.NE.2HFI) GOTO 406
       GOTO 4061
C
   406 IF (ICS(NPS).GT.0) GO TO 407
C
C      REMOVE AN APPLIED INTEGER CONSTRAINT
C      FROM THE STACK
C
       NPS=NPS-1
       IF (NPS.GT.0) GO TO 406
```

```
C
C     CREATE THE DISC FILE OUTINF
C
C     THE FOLLOWING SECTION MUST BE ADJUSTED IF THE ARRAY
C     SIZES INVOLVED WERE MODIFIED.
C
 4061 RKBB=FCNB
      KBB1(3)=NRB
      KBB1(4)=INIT
      CALL EXEC(15,66,KBB1,128,OUTINF,0)
      CALL EXEC(15,66,KBB2,128,OUTINF,1)
      CALL EXEC(15,66,KBB3,48,OUTINF,2)
      CALL EXEC(15,66,AB1,128,OUTINF,3)
      CALL EXEC(15,66,AB2,128,OUTINF,4)
      CALL EXEC(15,66,AB3,128,OUTINF,5)
      CALL EXEC(15,66,AB4,128,OUTINF,6)
      CALL EXEC(15,66,AB5,88,OUTINF,7)
C
C       PASS CONTROL TO THE PRODUCTION
C       REPORT GENERATOR
C
      CALL EXEC(10,IRGEN)
C
C
C     MODIFY AN INTEGER CONSTRAINT
C
  407 ICS(NPS)=-ICS(NPS)
      KCS(NPS)=-KCS(NPS)
      CS(NPS)=CS(NPS)-FLOAT(KCS(NPS))
      GO TO 411
C
C     ADD A NEW INTEGER CONSTRAINT TO THE STACK
C
  410 NPS=NPS+1
      ICS(NPS)=KB(ISM)
      KCS(NPS)=1
      CS(NPS)=FLOAT(IFIX(A(ISM,NC1)))
  411 DO 412 I=1,NPS
      J=IABS(ICS(I))
  412 IPS(J)=I
      GOTO 200
  103 FORMAT(1H )
  104 FORMAT(10I4)
  105 FORMAT(5E12.4)
  118 FORMAT (" NEGATIVE RHS IN ROW",I3)
  119 FORMAT ("*****RUN ABORTED")
  120 FORMAT ("NO STARTING BASIC VARIABLE IN ROW",I3)
      END
```

```
C
      FUNCTION A(I,J)
C
C     ************************************************
C     *                FUNCTION  A                   *
C     ************************************************
C
C     PURPOSE : FUNCTION A CONVERTS THE REFERENCES TO DENSE
C     LP MATRIX COEFFICIENTS A , USED IN ORIGINAL LP PROGRAM,
C     INTO REFERENCES TO STRING OF NON-ZERO MATRIX COEFFICIENTS
C
      COMMON Q(3000),IDUM(4),LQ(300),IDUM2(4),IQ(3000)
      A=0.
      M=0
      IF(J.GT.1) M=LQ(J-1)
      K=LQ(J)+1
    1 K=K-1
      IF(K.EQ.M) RETURN
      IF(I-IQ(K)) 1,2,3
    2 A=Q(K)
    3 RETURN
      END
      END$
::
:JF,ST
:PR,FTN4,2,1,99
:LU,7,0
:PR,LOADR,2,7
RLIB
%MLIB,%CLIB,RDIRC,RFLPA,/E
:ST,P
:TY
**** LIST END ****
```

```
:JO
:PU,ST,LPM2
:CL
:ST,S,ST,10
FTN4
      PROGRAM LPM2
C
C
C     **************************************************
C     *                                                *
C     *                PROGRAM LPM2                    *
C     *                                                *
C     **************************************************
C
C        LPM2 IS THE PRODUCTION REPORT GENERATOR OF THE
C        SYSTEM. THE PRODUCTION REPORT BREAK-DOWN IS THE
C        FOLLOWING :
C                   1.  REGULAR MARKER ASSIGNMENTS
C                   2.  LIST OF REJECTED REGULAR MARKERS
C                   3.  LIST OF DUMMY MARKERS
C                   4.  PRODUCTION SUMMARY
C                   5.  LIST OF CUT-DOWNS
C                   6.  BALANCE OF FABRIC
C
C        THE FOLLWING ARRAYS ARE USED IN LPM2 . THE SIZES OF
C        ARRAYS ARE SPECIFIED IN AN ATTACHED TABLE
C
C                   1.  KK1(N6),KK2(N6),....,KK5(N6) : LIST OF
C                       ID OF REJECTED REGULAR MARKERS. IN A
C                       LATER PHASE KK1(N6) IS USED AS POINTER
C                       TO THE SIZES IN A CUT-DOWN GROUP
C
C                   2.  IDM(N7)   : AUXILIARY ARRAYS USED FOR
C                                   CUT-DOWNS ALLOCATION
C                   3.  JDM(N7)   :           "
C                   4.  KDM(N7)   :           "
C                   5.  MDM(N7)   :           "
C                   6.  MR(N6)    :           "
C
C        THE INPUT OF LPM2 IS FROM LP8 AND LPG. THE FOLLOWING
C        DISC FILES ARE USED :
C
C                   1.  GENINF  : CREATED BY LPG
C                   2.  OUTINF  :    "    BY LP8
C                   3.  ITMP1   :    "    BY LPG
C                   4.  ITMP2   :    "    BY LPG
C
```

```
C
C     ***************************************************
C     *                                                 *
C     *            DISC FILE DOCUMENTATION              *
C     *                                                 *
C     ***************************************************
C
C     THE READING AND WRITTING ON THE DISC, PROCEDURES AND THE
C     SPACE ORGANIZATION IN THE DISC FILES ON THE DISC ARE DEPENTB
C     ON THE ARRAY SIZES. HENCE ANY CHANGE ON THE CURRENT MEMRY
C     SET-UP MUST BE FOLLOWED BY THE PROPER ADJUSTMENT OF THESE
C     PROCEDURES AND FILES.
C
C     A:   GENINF
C
C          SPECIFICATIONS :          TYPE OF DATA : BINARY
C                                    LENGTH       : 13 SECTORS
C                                                   (CURRENT SET-UP)
C                                    OPERATION    : OVER-WRITTING
C                                                   NO DIRECTORY
C
C          DISC FILE GENINF HAS THE FOLLOWING STRUCTURE
C
C     1. IDISC(4+4N3) :  EQUIVALENCED TO
C        DME          :  DUMMY MARKER EFFICIENCY
C        IS           :  NUMBER OF SIZES IN REQ. FILE
C        IPG          :  NUMBER OF FABRIC WIDTHS
C        PGW(N3)      :  FABRIC WIDTHS
C        PGQ(N3)      :  FABRIC QUANTITIES IN YDS ON STOCK
C                         ( FOR EVERY WIDTH )
C                       RESERVED SPACE : 1 SECTOR
C
C     2. JW(N2)       :  IDENTIFIES WIDTH FOR REAL MARKERS
C                         ( JW(J)=I, J-TH MARKER MADE FOR
C                            I-TH WIDTH )
C                       RESERVED SPACE : 1 SECTOR
C
C     3. AR(N1)        :  PATTERN AREA (FOR EVERY SIZE)
C                       RESERVED SPACE : 2 SECTORS
C
C     4. MKID1(N2),MKID2(N2),MKID3(N2),MKID4(N2),
C        MKID5(N2)    :  MARKER ID . TWO CHARACTERS PER WORD
C                       RESERVED SPACE : 5 SECTORS
C
C     ***********************************************************
```

```
C
C      B:    OUTINF
C
C            SPECIFICATIONS :          TYPE OF DATA : BINARY
C                                      LENGTH       : 8 SECTORS
C                                                     (CURRENT SET-UP)
C                                      OPERATION    : OVER-WRITTING
C                                                     NO DIRECTORY
C
C            DISC FILE OUTINF HAS THE FOLLOWING STRUCTURE
C
C      1. KBB(N5)      :  POINTER LINKING ACTUAL COLUMN INDECES
C                         IN LP SOLUTION WITH ORIGINAL INDECES OF
C                         UKNOWN VARIABLES. IT IS EQUIVALENCED TO
C                         KBB1,KBB2,KBB3, FROM BELOW , WHERE THE
C                         FIRST 3 WORDS OF KBB1 ARE EQUIVALENCED TO
C             NRB : NUMBER OF MARKERS ( INCLUDING DUMMY )
C             TAT : VALUE OF OBJECTIVE FUNCTION ( TOTAL FABRIC AREA)
C             INIT: NUMBER OF INTEGER ITERATION STEPS. PRINTED IN
C                   PRODUCTION REPORT AS "... REAL AND INT SOLUTIONS
C                        RESERVED SPACE : 3 SECTORS
C
C      2. AB(N5)       :  RESULTING VALUES OF UNKNOWN VARIABLES
C                          EQUIVALENCED TO : AB1,AB2,AB3,AB4,AB5
C                        RESERVED SPACE : 5 SECTORS
C
C      ***************************************************************
C
C      C.    ITMP1
C
C             SPECIFICATIONS :          TYPE OF DATA : BINARY
C                                       LENGTH       : 2+(N2/2) SCTS
C                                       OPERATION    : OVER-WRITTING
C                                                      NO DIRECTORY
C
C            DISC FILE ITMP1 HAS THE FOLLOWING STRUCTURE
C
C      1. RR(N5)       :  QUANTITIES OF GARMENTS REQUIRED
C                         ( RIGHT HAND SIDE OF THE MODEL )
C                         EQUIVALENCED TO RR1,RR2
C                        RESERVED SPACE : 2 SECTORS
C
C      2. IAWRK(N2)    :  WORKING AREA FOR STORING ROWISE THE
C                        MATRIX MA (IA FOR LPG), WHERE
C            MA(N1,N2) :  COEFFICIENTS OF THE PROBLEM
C                        RESERVED SPACE : (N2/2) SECTORS
C
C      ****************************************************************
C
```

```
C
      INTEGER OUTINF,GENINF
      DIMENSION CC(300),RR(128),MR(300),KC(128),IDM(128),
     &MA(60,100),AR(100),MKID1(100),MKID2(100),MKID3(100),
     &MKID4(100),MKID5(100),KK1(128),KK2(128),KK3(128),KK4(128),
     &KK5(128),JDM(128),PGW(10),PGQ(10),KBB(300),AB(300),KDM(128),
     &MDM(128),IDISC(44),JW(128),AR2(36),IDISC2(3),CC2(64),RR2(64)
     &IAWRK(120),GENINF(3),OUTINF(3),ITMP1(3),ITMP2(3),KBB1(128),
     &KBB2(128),KBB3(48),AB1(64),AB2(64),AB3(64),AB4(64),AB5(44)
      DIMENSION MONTR(3),EF(128)
C
      EQUIVALENCE (IDISC(1),DME),(IDISC(3),IS),(IDISC(4),IPG),
     &                 (IDISC(5),PGW),(IDISC(25),PGQ)
      EQUIVALENCE (AR(65),AR2),(IDISC2(1),NR),(IDISC2(2),NV),
     &                 (IDISC2(3),NRM)
      EQUIVALENCE (CC(65),CC2),(RR(65),RR2)
      EQUIVALENCE (KBB1(1),TAT),(KBB1(3),NRB),(KBB1(4),INIT),
     &                 (KBB1(128),KBB(124)),(KBB2(128),KBB(252)),
     &                  (KBB3(48),KBB(300))
      EQUIVALENCE (AB(1),AB1),(AB(65),AB2),(AB(129),AB3),
     &                 (AB(193),AB4),(AB(257),AB5)
C
      DATA OUTINF,GENINF,ITMP1,ITMP2/2HOU,2HTI,2HNF,2HGE,2HNI,
     &                  2HNF,2HIT,2HMP,2H1 ,2HIT,2HMP,2H2 /
      DATA MONTR/2HCO,2HR ,2H  /
      DATA LO/1/
C
C     **********      INPUT SECTION      **********
C
C     THIS SECTION MUST BE ADJUSTED PROPERLY IF THE ARRAY SIZES
C     INVOLVED WERE MODIFIED.
C
C
C     -1-             READ THE DISC FILE GENINF
C
      CALL EXEC(14,66,IDISC,44,GENINF,0)
      CALL EXEC(14,66,JW,128,GENINF,1)
      CALL EXEC(14,66,AR,128,GENINF,2)
      IF(NR.GT.64) CALL EXEC(14,66,AR2,72,GENINF,3)
      CALL EXEC(14,66,MKID1,100,GENINF,7)
      CALL EXEC(14,66,MKID2,100,GENINF,8)
      CALL EXEC(14,66,MKID3,100,GENINF,9)
      CALL EXEC(14,66,MKID4,100,GENINF,10)
      CALL EXEC(14,66,MKID5,100,GENINF,11)
      CALL EXEC(14,66,EF,256,GENINF,12)
C
C     -2-             READ THE DISC FILE ITMP2
C
      CALL EXEC(14,66,IDISC2,3,ITMP2,0)
      CALL EXEC(14,66,CC,128,ITMP2,2)
      IF(NV.GT.64) CALL EXEC(14,66,CC2,128,ITMP2,3)
```

```
C
C     -3-               READ THE DISC FILE ITMP1
C
      NMK=NV-IS
      CALL EXEC(14,66,RR,128,ITMP1,0)
      IF(IS.GT.64) CALL EXEC(14,66,RR2,128,ITMP1,1)
      ISECT=2
      DO 410 K=1,99,2
      CALL EXEC(14,66,IAWRK,120,ITMP1,ISECT)
      DO 400 I=1,60
      MA(I,K)=IAWRK(I)
  400 MA(I,K+1)=IAWRK(I+60)
      ISECT=ISECT+1
  410 CONTINUE
      CALL EXEC(14,66,KC,128,ITMP1,ISECT)
C
C     -4-               READ THE DISC FILE OUTINF
C
      CALL EXEC(14,66,KBB1,128,OUTINF,0)
      CALL EXEC(14,66,KBB2,128,OUTINF,1)
      CALL EXEC(14,66,KBB3,48,OUTINF,2)
      CALL EXEC(14,66,AB1,128,OUTINF,3)
      CALL EXEC(14,66,AB2,128,OUTINF,4)
      CALL EXEC(14,66,AB3,128,OUTINF,5)
      CALL EXEC(14,66,AB4,128,OUTINF,6)
      CALL EXEC(14,66,AB5,88,OUTINF,7)
C
C     ***********************************************
C
      WRITE(LO,5) INIT
C
C
      TMG=RR(1)
      TAP=RR(1)*AR(1)
      ICD=0
      DO 441 I=2,NR
      IF(KC(I).EQ.1) ICD=1
      TMG=TMG+RR(I)
  441 TAP=TAP+RR(I)*AR(I)
```

```
C
C      PRINTING OF REPORT HEADING AND REGULAR
C      MARKER ASSIGNMENTS
C
       WRITE(LO,15Ø) KD,NAME
       WRITE(LO,6)
       WRITE(LO,155)
       NMKS=Ø
       DO 45Ø I=1,NRB
       IF(KBB(I).GT.NMK) GOTO 4511
       IF(AB(I).EQ.Ø.) GOTO 45Ø
       NMKS=NMKS+1
       WRITE(LO,129) NMKS,MKID1(KBB(I)),MKID2(KBB(I)),MKID3(KBB(I))
      &             MKID4(KBB(I)),MKID5(KBB(I)),AB(I),EF(KBB(I))
  45Ø  CONTINUE
  4511 JK=1
       K=Ø
       DO 449 J=1,NMK
       IF(KBB(JK).EQ.J) GO TO 448
  447  K=K+1
       KK1(K)=MKID1(J)
       KK2(K)=MKID2(J)
       KK3(K)=MKID3(J)
       KK4(K)=MKID4(J)
       KK5(K)=MKID5(J)
       GOTO 449
  448  JK=JK+1
       IF(AB(JK-1).EQ.Ø.) GOTO 447
  449  CONTINUE
       IF(K.EQ.Ø) GOTO 451
C
C      PRINTING THE LIST OF REJECTED REGULAR MARKERS
C
       WRITE(LO,156)
       WRITE(LO,158) (KK1(J),KK2(J),KK3(J),KK4(J),KK5(J),J=1,K)
  451  WRITE(LO,1Ø3)
```

```
C
C      PRINT THE LIST OF DUMMY MARKERS
C
       WRITE(LO,157)
       IA=I
       IJ=0
       DAT=0.
       DMG=0.
       DO 452 I=IA,NRB
       IF(KBB(I).GT.NV) GOTO 453
       IF(AB(I).EQ.0.) GOTO 452
       IJ=IJ+1
       DAT=DAT+AB(I)*CC(KBB(I))
       DMG=DMG+AB(I)
       IR=KBB(I)-NMK
       RR(IR)=RR(IR)-AB(I)
       WRITE(LO,130) IJ,MKID1(KBB(I)),MKID2(KBB(I)),MKID3(KBB(I)),
      &               AB(I)
  452  CONTINUE
C
C      EVALUATE THE DATA AND PRINT THE SUMMARY
C
  453  DAP=DAT*DME
       RAP=TAP-DAP
       RAT=TAT-DAT
       DME=100.*DME
       RME=100.*RAP/RAT
       TME=100.*TAP/TAT
       RMG=TMG-DMG
       RMP=100.*RMG/TMG
       DMP=100.-RMP
       WRITE(LO,151)
       WRITE(LO,152)
       WRITE(LO,153)
       WRITE(LO,154) RMG,RMP,RAT,RAP,RME
       WRITE(LO,153)
       WRITE(LO,170) DMG,DMP,DAT,DAP,DME
       WRITE(LO,153)
       WRITE(LO,152)
       WRITE(LO,153)
       WRITE(LO,171) TMG,100.,TAT,TAP,TME
       WRITE(LO,153)
       IF(ICD.EQ.0)  GOTO 100
```

```
C
C      ALLOCATE AND PRINT THE LIST OF CUT-DOWNS
C
       NDM=Ø
       I=Ø
 503   I=I+1
       IF(I.GT.IS) GOTO 512
       MR(I)=IFIX(RR(I)+.ØØØØ1)
       IF(KC(I).EQ.Ø) GOTO 5Ø3
       IA=I
 5Ø4   I=I+1
       MR(I)=IFIX(RR(I)+.ØØØØ1)
       IF(KC(I).EQ.1) GOTO 5Ø4
       K=Ø
       DO 5Ø6 II=IA,I
       K=K+1
       KK1(K)=II
       J=Ø
 5Ø5   J=J+1
       MA(K,J)=MA(II,KBB(J))*IFIX(AB(J)+.ØØØØ1)
       IF(KBB(J+1).LE.NMK) GOTO 5Ø5
 5Ø6   CONTINUE
       DO 5Ø7 II=1,K
       DO 5Ø7 JJ=1,J
 5Ø7   MR(KK1(II))=MR(KK1(II))-MA(II,JJ)
 5Ø8   MRM=Ø
       DO 5Ø9 II=1,K
       IF(MRM.GT.MR(KK1(II))) GOTO 5Ø9
       IMRM=KK1(II)
       MRM=MR(IMRM)
       IA=II+1
 5Ø9   CONTINUE
       IF(MRM.EQ.Ø) GOTO 5Ø3
       MM=Ø
       DO 51Ø II=IA,K
       DO 51Ø JJ=1,J
       IF(MA(II,JJ).EQ.MR(IMRM).AND.MR(KK1(II))+MR(IMRM).
      &LE.Ø) GOTO 511
       MMM=MINØ(MA(II,JJ),-MR(KK1(II)))
       IF(MM.GT.MMM) GOTO 51Ø
       MM=MMM
       IMM=II
       JMM=JJ
 51Ø   CONTINUE
```

```
      MCTD=MIN0(MRM,MM)
      MR(IMRM)=MR(IMRM)-MCTD
      MR(KK1(IMM))=MR(KK1(IMM))+MCTD
      MA(IMM,JMM)=MA(IMM,JMM)-MCTD
      NDM=NDM+1
      IDM(NDM)=KBB(JMM)
      JDM(NDM)=KK1(IMM)
      KDM(NDM)=IMRM
      MDM(NDM)=MCTD
      GOTO 508
511   MR(IMRM)=0
      MR(KK1(II))=MR(KK1(II))+MR(IMRM)
      MA(II,JJ)=0
      NDM=NDM+1
      IDM(NDM)=KBB(JJ)
      JDM(NDM)=KK1(II)
      KDM(NDM)=IMRM
      MDM(NDM)=MRM
      GOTO 508
512   IF(NDM.EQ.0) GOTO 100
      WRITE(LO,159)
      DO 513 I=1,NMK
      MIDP1=MKID1(KBB(I))
      MIDP2=MKID2(KBB(I))
      MIDP3=MKID3(KBB(I))
      MIDP4=MKID4(KBB(I))
      MIDP5=MKID5(KBB(I))
      DO 513 J=1,NDM
      IF(IDM(J).NE.KBB(I)) GOTO 513
      JDMJ=JDM(J)+NMK
      KDMJ=KDM(J)+NMK
      WRITE(LO,160) MIDP1,MIDP2,MIDP3,MIDP4,MIDP5,MKID1(JDMJ),
     &MKID2(JDMJ),MKID3(JDMJ),MKID1(KDMJ),MKID2(KDMJ),MKID3(KDMJ),
     &MDM(J)
      MIDP1=2H
      MIDP2=2H-
      MIDP3=2H"
      MIDP4=2H-
      MIDP5=2H
513   CONTINUE
100   IF(IPG.EQ.0) GOTO 200
```

```
C
C      EVALUATE AND PRINT THE BALANCE OF FABRIC
C      WIDTHS AND QUANTITIES
C
       WRITE(LO,161)
       DO 522 I=1,IPG
       ASA=0.
       DO 521 J=1,NMK
       IF(JW(KBB(J)).EQ.I) ASA=ASA+AB(J)*CC(KBB(J))
  521  CONTINUE
       PGQ2=PGQ(I)*PGW(I)/36.
       ASL=36.*ASA/PGW(I)
       RL=PGQ(I)-ASL
       RA=PGQ2-ASA
       WRITE(LO,162) PGW(I),PGQ(I),PGQ2,ASL,ASA,RL,RA
  522  CONTINUE
  200  WRITE(LO,103)
C
C
C      RETURN CONTROL TO MONITOR
C
       CALL EXEC(10,MONTR)
C
C      **********      FORMAT  TABLE      **********
C
C
    5 FORMAT(1X,I4,"  REAL AND INT SOLUTIONS")
    6 FORMAT("  REGULAR MARKERS")
    7 FORMAT(1X,4(I4,1X),E10.5,2X,I4)
    8 FORMAT(1X,5(I4,2X))
  103 FORMAT(1H )
  129 FORMAT(6X,I6,6X,5A2,2X,F6.0,F8.2)
  130 FORMAT(6X,I6,6X,3A2,6X,F6.0)
  150 FORMAT(//20X,"MARKAMATIC CUT ORDER PLANNING"/20X,"---------",
     &"---------------------"/19X,5A2,5X,"RQ FILE: ",5A2/)
  151 FORMAT(//" S U M M A R Y :"//"I",17X,"I",5X,"COVERAGE",4X,
     &"I",3X,"TOTAL",2X,"I",2X,"PATTERN",1X,"I",1X,"EFFIC",1X,
     &"I"/"I",17X,"I",2X,"GARMTS",1X,"I",3X,"%",3X,"I",3X,"AREA",
     &3X,"I",3X,"AREA",3X,"I",3X,"%",3X,"I")
  152 FORMAT("I",17(1H-),"I",9(1H-),"I",7(1H-),"I",10(1H-),"I",
     &10(1H-),"I",7(1H-),"I")
  153 FORMAT("I",17X,"I",9X,"I",7X,"I",10X,"I",10X,"I",7X,"I")
  154 FORMAT("I REGULAR MARKERS I",F7.0,"  I ",F5.1," I",F8.2,
     &"  I",F8.2,"  I ",F5.2," I")
  155 FORMAT(/4X,"A - ACCEPTED"/18X,"MARKER ID",5X,"PLIES",2X,
     &"EFF %"/9X,35(1H-))
  158 FORMAT(18X,5A2,2X,5A2,2X,5A2,2X,5A2)
  160 FORMAT(13X,5A2,7X,3A2,3X,3A2,2X,I5,".")
```

```
      159 FORMAT(//" CUT-DOWNS:"/13X,"MARKER ID    FROM SIZE",
         &  "  TO SIZE      PLIES"/
         &12X,"------------------------------------------")
      156 FORMAT(/4X,"B - REJECTED")
      157 FORMAT(/" DUMMY  MARKERS:"/18X,"SIZE",1ØX,"GMTS"/
         &9X,"-------------------------------")
      161 FORMAT(//" FABRIC BALANCE:"/
         &7X,"WIDTH         AVAILABLE      ASSIGNED TO RM        ",
         &"REMAINS"/
         &8X,"IN          YD       SQYD        YD      SQYD        YD",
         &"       SQYD"/
         &5X,58(1H-))
      162 FORMAT(F12.2,1X,3(F9.1,F8.1))
      17Ø FORMAT("I    DUMMY MARKERS I",F7.Ø,"  I ",F5.1," I",F8.2,
         &"  I",F8.2,"  I ",F5.2," I")
      171 FORMAT("I",5X,"T O T A L",3X,"I",F7.Ø,"  I ",F5.1," I",F8.2,
         &"  I",F8.2,"  I ",F5.2," I")
    C
    C
          END
          END$
    ::
    :JF,ST
    :PR,FTN4,2,1,99
    :LU,7,Ø
    :PR,LOADR,2,7
    RLIB
    %MLIB,%CLIB,/E
    :ST,P
    :TY
**** LIST END ****
```

Requirement files, marker banks

an production reports

Examples

Notes: DME . . . . dummy marker efficiency

The results from Cyber and HP are almost identical (only a few numbers differ in last digit).

CPU time is indicated on the output from both Cyber and HP. The HP runs took approximately 10 times more time (scores of seconds for the larger problem OXREQN/OXMBN).

```
/C,OXREQT
CT OXREQT
ST DUA
SZ 2934,19
SZ 3132,14
SZ 3229,22
SZ 3230,29,Y
SZ 3231,12,Y
SZ 3234,10,Y
SZ 3331,18
SZ 3333,33,Y
SZ 3427,11
SZ 3429,19,Y
SZ 3430,25,Y
SZ 3431,12,Y
SZ 3432,21,Y
SZ 3433,16,Y
SZ 3434,17,Y
SZ 3628,15
SZ 3632,16,Y
SZ 3634,17,Y
SZ 3729,18
SZ 3734,23,Y
SZ 3930,5
PG 58,100
PG 59,150
PG 60,120
/
```

/C,OXMBSS
2926,1.4389
2927,1.4708
2928,1.5026
2929,1.5344
2930,1.5663
2931,1.5981
2932,1.6300
2933,1.6618
2934,1.6937
3026,1.4728
3027,1.5047
3028,1.5365
3029,1.5684
3030,1.6002
3031,1.6321
3032,1.6639
3033,1.6957
3034,1.7276
3126,1.5068
3127,1.5386
3128,1.5704
3129,1.6023
3130,1.6341
3131,1.6660
3132,1.6978
3133,1.7297
3134,1.7615
3226,1.5407
3227,1.5725
3228,1.6044
3229,1.6362
3230,1.6681
3231,1.6999
3232,1.7317
3233,1.7636
3234,1.7954
3326,1.5746
3327,1.6065
3328,1.6383
3329,1.6701
3330,1.7020
3331,1.7338
3332,1.7657
3333,1.7975
3334,1.8294
3426,1.6085
3427,1.6404
3428,1.6722
3429,1.7041
3430,1.7359
3431,1.7678
3432,1.7996
3433,1.8314
3434,1.8633
3526,1.6425
3527,1.6743
3528,1.7061
3529,1.7380
3530,1.7698
3531,1.8017
3532,1.8335
3533,1.8654
3534,1.8972
3626,1.6764
3627,1.7082
3628,1.7401
3629,1.7719
3630,1.8038
3631,1.8356
3632,1.8674
3633,1.8993
3634,1.9311
3726,1.7103
3727,1.7422
3728,1.7740
3729,1.8058
3730,1.8377
3731,1.8695
3732,1.9014
3733,1.9332
3734,1.9651
3826,1.7442
3827,1.7761
3828,1.8079
3829,1.8398
3830,1.8716
3831,1.9035
3832,1.9353
3833,1.9671
3834,1.9990
3926,1.7782
3927,1.8100
3928,1.8418
3929,1.8737
3930,1.9055
3931,1.9374
3932,1.9692
3933,2.0011
3934,2.0329
/

```
DUA5800201,DUA,58, 2:54
3432,1
3632,1
/END
DUA5900202,DUA,59, 2:32
3329,1
3627,1
/END
DUA6000203,DUA,59, 2:97
3632,1
3828,1
/END
DUA5900206,DUA,58,2:52
3430,2
/END
DUA5900207,DUA,59, 2.60
3434,2
/END
DUA5900211,DUA,60,2:60
3532,1
3828,1
/END
DUA5900215,DUA,58,2:56
3430,1
3432,1
/END
DUA5900225,DUA,59, 2:62
3232,1
3930,1
/END
DUA5900228,DUA,60,2:42
3232,2
/END
DUA5900231,DUA,58,2:64
3434,1
3532,1
/END
DUA5900233,DUA,59, 2.68
3330,1
3930,1
/END
DUA5900234,DUA,60,2:54
3628,1
3732,1
/END
DUA5900236,DUA,58,2:50,87.68
3428,1
3630,1
/END
DUA5900243,DUA,59, 2:44
3428,1
3430,1
/END
DUA5900244,DUA,60,2.54
3230,1
3928,1
/END
DUA5900248,DUA,58,2:73
3332,1
3832,1
/END
DUA5900704,DUA,59, 9:82
3630,1
3630,1
3930,1
4030,1
4232,1
4628,1
4628,1
/END
DUA5900721,DUA,60,9:72
4028,1
4228,1
4230,1
4428,1
4432,1
4628,1
4632,1
/END
DUA5901004,DUA,58,13:20
3232,1
3330,1
3632,1
3634,1
3732,1
3828,1
3832,1
3928,1
3934,1
4034,1
/END
```

```
/RUNLPM2
  REQUIREMENT FILE? OXREQT
  MARKER BANK ?-OXMBGG
  DUMMY MARKER EFFICIENCY ? .82
  FIRST-BEST INTEGER SOLUTION ? FIRST
 4 REAL AND INT SOLUTIONS
```

MARKAMATIC CUT ORDER PLANNING

76/12/13.   RQ FILE: OXREQT

REGULAR MARKERS:

A - ACCEPTED

	MARKER ID	PLIES	EFF %
1	DUA5900201	16.	89.26
2	DUA5900205	12.	87.02
3	DUA5900206	6.	87.08
4	DUA5900207	4.	84.10
5	DUA5900210	13.	85.62
6	DUA5900213	19.	84.89
7	DUA5900215	2.	85.61
8	DUA5900217	18.	86.22
9	DUA5900232	1.	83.97
10	DUA5900235	17.	82.96
11	DUA5900240	2.	83.52
12	DUA5900243	19.	84.59
13	DUA5900246	4.	85.31

DUMMY MARKERS:

	SIZE	GMTS
1	3229	5.
2	3230	23.
3	3333	32.
4	3427	11.
5	3434	17.
6	3729	18.
7	3930	5.

S U M M A R Y :

	COVERAGE GARMTS	COVERAGE %	TOTAL AREA	PATTERN AREA	EFFIC %
REGULAR MARKERS	266.	70.6	552.08	472.06	85.51
DUMMY MARKERS	111.	29.4	238.81	195.82	82.00
T O T A L	377.	100.0	790.89	667.88	84.45

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900205	3432	3431	5.
- " -	3432	3430	4.
DUA5900217	3231	3230	6.

FABRIC BALANCE:

WIDTH IN	AVAILABLE YD	AVAILABLE SQYD	ASSIGNED TO RM YD	ASSIGNED TO RM SQYD	REMAINS YD	REMAINS SQYD
58.00	100.0	161.1	97.6	157.2	2.4	3.9
59.00	150.0	245.8	133.1	218.2	16.9	27.6
60.00	120.0	200.0	106.0	176.6	14.0	23.4

7.075 CP SECONDS EXECUTION TIME

```
:PR, COR
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$M
CT  OXREQ3
  DUMMY MARKER EFFICIENCY :
.82
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$E
   FIRST-BEST INTEGER SOLUTION
FIRST
    4  REAL AND INT SOLUTIONS
```

65 seconds

MARKAMATIC CUT ORDER PLANNING

REGULAR MARKERS

A - ACCEPTED

	MARKER ID	PLIES	EFF %
1	DUA59ØØ2Ø1	16.	89.26
2	DUA59ØØ2Ø5	12.	87.Ø2
3	DUA59ØØ2Ø6	6.	87.Ø8
4	DUA59ØØ2Ø7	4.	84.1Ø
5	DUA59ØØ21Ø	13.	85.62
6	DUA59ØØ213	19.	84.89
7	DUA59ØØ215	2.	85.61
8	DUA59ØØ217	18.	86.22
9	DUA59ØØ232	1.	83.97
1Ø	DUA59ØØ235	17.	82.96
11	DUA59ØØ24Ø	2.	83.52
12	DUA59ØØ243	19.	84.59
13	DUA59ØØ246	4.	85.31

DUMMY MARKERS:

	SIZE	GMTS
1	3229	5.
2	3230	23.
3	3333	32.
4	3427	11.
5	3434	17.
6	3729	18.
7	3930	5.

S U M M A R Y :

	COVERAGE GARMTS	COVERAGE %	TOTAL AREA	PATTERN AREA	EFFIC %
REGULAR MARKERS	266.	70.6	552.05	472.06	85.51
DUMMY MARKERS	111.	29.4	238.80	195.82	82.00
T O T A L	377.	100.0	790.86	667.88	84.45

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900205	3432	3431	5.
- " -	3432	3430	4.
DUA5900217	3231	3230	6.

FABRIC BALANCE:

WIDTH IN	AVAILABLE YD	AVAILABLE SQYD	ASSIGNED TO RM YD	ASSIGNED TO RM SQYD	REMAINS YD	REMAINS SQYD
58.00	100.0	161.1	97.6	157.2	2.4	3.9
59.00	150.0	245.8	133.1	218.2	16.9	27.6
60.00	120.0	200.0	106.0	176.6	14.0	23.4

```
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$S
  COR   : STOP    0000

@
```

```
/RUNLPM2
  REQUIREMENT FILE? OXREQT
  MARKER BANK ? OXMBSS
  DUMMY MARKER EFFICIENCY ? .83
  FIRST-BEST INTEGER SOLUTION ? FIRST
 3 REAL AND INT SOLUTIONS
```

MARKAMATIC CUT ORDER PLANNING

76/12/13.   RQ FILE: OXREQT

REGULAR MARKERS:

A - ACCEPTED

	MARKER ID	PLIES	EFF %
1	DUA5900201	16.	89.26
2	DUA5900205	12.	87.02
3	DUA5900206	8.	87.08
4	DUA5900207	3.	84.10
5	DUA5900210	14.	85.62
6	DUA5900213	19.	84.89
7	DUA5900217	18.	86.22
8	DUA5900240	1.	83.52
9	DUA5900243	19.	84.59
10	DUA5900246	4.	85.31

B - REJECTED

DUA5900215   DUA5900232

DUMMY MARKERS:

	SIZE	GMTS
1	3229	22.
2	3230	23.
3	3234	3.
4	3333	33.
5	3427	11.
6	3434	17.
7	3634	17.
8	3729	18.
9	3930	5.

S U M M A R Y :

	COVERAGE GARMTS	COVERAGE %	TOTAL AREA	PATTERN AREA	EFFIC %
REGULAR MARKERS	228.	60.5	470.19	404.23	85.97
DUMMY MARKERS	149.	39.5	317.65	263.65	83.00
T O T A L	377.	100.0	787.84	667.88	84.77

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900205	3432	3430	6.
- " -	3432	3431	1.
DUA5900217	3231	3230	6.

FABRIC BALANCE:

WIDTH IN	AVAILABLE YD	AVAILABLE SQYD	ASSIGNED TO RM YD	ASSIGNED TO RM SQYD	REMAINS YD	REMAINS SQYD
58.00	100.0	161.1	100.0	161.1	.0	.0
59.00	150.0	245.8	128.0	209.8	22.0	36.0
60.00	120.0	200.0	59.6	99.3	60.4	100.7

4.322 CP SECONDS EXECUTION TIME

```
 :PR,COR
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$M
CT  OXREQ3
  DUMMY MARKER EFFICIENCY :
.83
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$E
   FIRST-BEST INTEGER SOLUTION
FIRST
    3  REAL AND INT SOLUTIONS
```

55 seconds

MARKAMATIC CUT ORDER PLANNING

REGULAR MARKERS

A - ACCEPTED

	MARKER ID	PLIES	EFF %
1	DUA5900201	16.	89.26
2	DUA5900205	12.	87.02
3	DUA5900206	8.	87.08
4	DUA5900207	3.	84.10
5	DUA5900210	14.	85.62
6	DUA5900213	19.	84.89
7	DUA5900217	18.	86.22
8	DUA5900240	1.	83.52
9	DUA5900243	19.	84.59
10	DUA5900246	4.	85.31

B - REJECTED

DUA5900215 DUA5900232 DUA5900235

DUMMY MARKERS:

	SIZE	GMTS
1	3229	22.
2	3230	23.
3	3234	3.
4	3333	33.
5	3427	11.
6	3434	17.
7	3634	17.
8	3729	18.
9	3930	5.

S U M M A R Y :

	COVERAGE GARMTS	COVERAGE %	TOTAL AREA	PATTERN AREA	EFFIC %
REGULAR MARKERS	228.	60.5	470.21	404.23	85.97
DUMMY MARKERS	149.	39.5	317.65	263.65	83.00
T O T A L	377.	100.0	787.86	667.88	84.77

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900205	3432	3430	6.
- " -	3432	3431	1.
DUA5900217	3231	3230	6.

FABRIC BALANCE:

WIDTH	AVAILABLE		ASSIGNED TO RM		REMAINS	
IN	YD	SQYD	YD	SQYD	YD	SQYD
58.00	100.0	161.1	100.0	161.1	.0	.0
59.00	150.0	245.8	128.0	209.8	22.0	36.0
60.00	120.0	200.0	59.6	99.3	60.4	100.7

ENTER INPUT DEVICE #
1
ENTER CONTROL CARD
$S
COR : STOP 0000

@

```
/C,OXREQN
OXREQN
ST DUA
SZ 3230,50
SZ 3232,70,Y
SZ 3330,60
SZ 3332,80,Y
SZ 3428,50
SZ 3430,70,Y
SZ 3432,80,Y
SZ 3434,90,Y
SZ 3528,60
SZ 3530,70,Y
SZ 3532,80,Y
SZ 3628,70
SZ 3630,80,Y
SZ 3632,70,Y
SZ 3634,60,Y
SZ 3728,100
SZ 3730,60,Y
SZ 3732,90,Y
SZ 3826,60
SZ 3830,70,Y
SZ 3832,100,Y
SZ 3928,70
SZ 3930,90,Y
SZ 3932,100,Y
SZ 3934,110,Y
SZ 4028,60
SZ 4030,70,Y
SZ 4032,80,Y
SZ 4034,90,Y
SZ 4132,70
SZ 4134,80,Y
SZ 4228,80
SZ 4230,90,Y
SZ 4232,110,Y
SZ 4234,50,Y
SZ 4328,70
SZ 4330,40,Y
SZ 4332,50,Y
SZ 4428,60
SZ 4430,50,Y
SZ 4432,55,Y
SZ 4434,70,Y
SZ 4626,70
SZ 4628,80,Y
SZ 4630,50,Y
SZ 4632,60,Y
SZ 4634,70,Y
PG 58,3000
PG 59,4000
PG 60,5000
/
```

/C,OXMBN
3230,1.6681
3232,1.7317
3330,1.7020
3331,1.7338
3332,1.7657
3333,1.7975
3334,1.8294
3426,1.6085
3427,1.6404
3428,1.6722
3429,1.7041
3430,1.7359
3431,1.7678
3432,1.7996
3433,1.8314
3434,1.8633
3526,1.6425
3527,1.6743
3528,1.7061
3529,1.7380
3530,1.7698
3531,1.8017
3532,1.8335
3533,1.8654
3534,1.8972
3626,1.6764
3627,1.7082
3628,1.7401
3629,1.7719
3630,1.8038
3631,1.8356
3632,1.8674
3633,1.8993
3634,1.9311
3726,1.7103
3727,1.7422
3728,1.7740
3729,1.8058
3730,1.8377
3731,1.8695
3732,1.9014
3733,1.9332
3734,1.9651
3826,1.7442
3827,1.7761
3828,1.8079
3829,1.8398
3830,1.8716
3831,1.9035
3832,1.9353
3833,1.9671
3834,1.9990
3926,1.7782
3927,1.8100
3928,1.8418
3929,1.8737
3930,1.9055
3931,1.9374
3932,1.9692
3933,2.0011
3934,2.0329
4026,1.8121
4027,1.8439
4028,1.8758
4029,1.9076
4030,1.9395
4031,1.9713
4032,2.0031
4033,2.0350
4034,2.0668
4126,1.8460
4127,1.8779
4128,1.9097
4129,1.9415
4130,1.9734
4131,2.0052
4132,2.0371
4133,2.0689
4134,2.1008
4226,1.8799
4227,1.9118
4228,1.9436
4229,1.9755
4230,2.0073
4231,2.0392
4232,2.0710
4233,2.1028
4234,2.1347
4326,1.9139
4327,1.9457
4328,1.9775
4329,2.0094
4330,2.0412
4331,2.0731
4332,2.1049
4333,2.1368
4334,2.1686
4426,1.9478
4427,1.9796
4428,2.0115
4429,2.0433
4430,2.0752
4431,2.1070
4432,2.1388
4433,2.1707
4434,2.2025
4526,1.9817
4527,2.0136
4528,2.0454
4529,2.0772
4530,2.1091
4531,2.1409
4532,2.1728
4533,2.2046
4534,2.2365
4626,2.0156
4627,2.0475
4628,2.0793
4629,2.1112
4630,2.1430
4631,2.1749
4632,2.2067
4633,2.2385
4634,2.2704
/

```
DUA5900201,DUA,58
3432,1
3632,1
/END
DUA5900202,DUA,58
3329,1
3627,1
/END
DUA5900203,DUA,58
3633,1
3828,1
/END
DUA5900204,DUA,58
3030,1
3329,1
/END
DUA5900205,DUA,58
3432,1
3433,1
/END
DUA5900206,DUA,58
3431,2
/END
DUA5900207,DUA,58
3234,2
/END
DUA5900208,DUA,58
3234,1
3528,1
/END
DUA5900209,DUA,58
3234,1
3528,1
/END
DUA5900210,DUA,59
3132,1
3628,1
/END
DUA5900211,DUA,59
3533,1
3829,1
/END
DUA5900212,DUA,59
3334,1
3626,1
/END
DUA5900213,DUA,59
2934,1
3734,1
/END
DUA5900214,DUA,59
3627,2
/END
DUA5900215,DUA,59
3430,1
3432,1
/END
DUA5900216,DUA,59
3234,1
3534,1
/END
DUA5900217,DUA,59
3231,1
3331,1
/END
DUA5900232,DUA,59
3132,1
3333,1
/END
DUA5900233,DUA,60
3330,1
3931,1
/END
DUA5900234,DUA,60
3629,1
3732,1
/END
DUA5900235,DUA,60
3229,1
3634,1
/END
DUA5900240,DUA,60
3234,1
3628,1
/END
DUA5900241,DUA,60
3432,1
3627,1
/END
DUA5900242,DUA,60
3530,2
/END
DUA5900243,DUA,60
3429,1
3430,1
/END
DUA5900244,DUA,60
3230,1
3929,1
/END
DUA5900245,DUA,60
3532,1
3534,1
/END
DUA5900246,DUA,60
3433,1
3734,1
/END
&
```

```
DUA5901005,DUA,59,12:88
3332,1
3430,1
3532,1
3628,1
3630,1
3632,1
3828,1
3830,1
3832,1
4034,1
/END
DUA5901102,DUA,60,14:50
3528,1
3530,1
3630,1
3632,1
3732,1
3830,1
3832,1
4232,1
4328,1
4626,1
4628,1
/END
DUA5901106,DUA,58,15:1
3230,1
3430,1
3430,1
3728,1
3732,1
4032,1
4130,1
4232,1
4330,1
4430,1
4432,1
/END
DUA5901108,DUA,60,14:53
3428,1
3628,1
3630,1
3634,1
3732,1
3734,1
3830,1
3834,1
4228,1
4234,1
4434,1
/END
DUA5901301,DUA,58,17:42,
3430,1
3430,1
3628,1
3630,1
3828,1
3830,1
3832,1
4028,1
4228,1
4230,1
4234,1
4428,1
4432,1
/END
DUA5901305,DUA,59,17:44
3432,1
3628,1
3630,1
3630,1
3432,1
3830,1
3830,1
3832,1
4030,1
4230,1
4232,1
4432,1
4632,1
/END
DUA5901308,DUA,60,17:58
3430,1
3630,1
3632,1
3830,1
3832,1
3928,1
4030,1
4032,1
4232,1
4330,1
4430,1
4434,1
4630,1
/END
```

```
DUA5901311,DUA,58,17:17
3428,1
3430,1
3528,1
3628,1
3630,1
3728,1
3828,1
4028,1
4030,1
4228,1
4428,1
4628,1
4632,1
/END
DUA5901312,DUA,59,17:32
3332,1
3434,1
3628,1
3630,1
3728,1
3826,1
3930,1
3934,1
4030,1
4132,1
4332,1
4428,1
4628,1
/END
DUA5901313,DUA,60,16:72
3232,1
3428,1
3528,1
3628,1
3728,1
3828,1
3930,1
4028,1
4130,1
4228,1
4332,1
4430,1
4632,1
/END
&
```

```
/RUNLPM2
  REQUIREMENT FILE? OXREQN
  MARKER BANK ? OXMBN
  DUMMY MARKER EFFICIENCY ? .82
  FIRST-BEST INTEGER SOLUTION ? FIRST
 2 REAL AND INT SOLUTIONS
```

MARKAMATIC CUT ORDER PLANNING

76/12/13. RQ FILE: OXREQN

REGULAR MARKERS:

A - ACCEPTED

	MARKER ID	PLIES	EFF %
1	DUA5900215	60.	85.72
2	DUA5900225	85.	84.71
3	DUA5900228	7.	85.87
4	DUA5900231	105.	86.92
5	DUA5900234	20.	86.02
6	DUA5900243	5.	85.23
7	DUA5900244	20.	82.91
8	DUA5900248	5.	84.15
9	DUA5900721	55.	88.04
10	DUA5901102	45.	86.89
11	DUA5901308	50.	87.14
12	DUA5901312	5.	87.37

B - REJECTED

DUA5800201	DUA5900206	DUA5900207	DUA5900233
DUA5900236	DUA5900704	DUA5901305	

DUMMY MARKERS:

	SIZE	GMTS
1	3230	1.
2	3330	60.
3	3332	70.
4	3528	15.
5	3634	60.
6	3728	95.
7	3730	60.
8	3732	25.
9	3826	30.
10	3932	100.
11	3934	105.
12	4028	5.
13	4030	15.
14	4032	30.
15	4034	90.
16	4132	65.
17	4134	80.
18	4228	25.
19	4230	35.
20	4232	15.
21	4234	50.
22	4328	15.
23	4332	45.
24	4434	20.
25	4632	5.
26	4634	70.

S U M M A R Y :

	COVERAGE		TOTAL	PATTERN	EFFIC
	GARMTS	%	AREA	AREA	%
REGULAR MARKERS	2209.	65.1	4884.04	4224.80	86.50
DUMMY MARKERS	1186.	34.9	2851.18	2337.97	82.00
T O T A L	3395.	100.0	7735.22	6562.77	84.84

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900225	3232	3230	29.
DUA5900231	3434	3432	20.
- " -	3532	3530	25.
DUA5901102	4628	4626	25.
DUA5901308	3430	3428	45.
- " -	3632	3628	25.
- " -	3630	3628	20.
- " -	3830	3826	25.
- " -	4330	4328	10.

FABRIC BALANCE:

WIDTH	AVAILABLE		ASSIGNED TO RM		REMAINS	
IN	YD	SQYD	YD	SQYD	YD	SQYD
58.00	3000.0	4833.3	444.4	716.1	2555.6	4117.3
59.00	4000.0	6555.6	321.5	526.9	3678.5	6028.6
60.00	5000.0	8333.3	2184.6	3641.1	2815.4	4692.3

11.605 CP SECONDS EXECUTION TIME

```
:PR, COR
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$M
CT  OXREQB
  DUMMY MARKER EFFICIENCY :
.82
  ENTER THE AREA OF THE SIZE : 3434
1.8633
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$E
   FIRST-BEST INTEGER SOLUTION
FIRST
    2  REAL AND INT SOLUTIONS
```

95 seconds

MARKAMATIC CUT ORDER PLANNING

REGULAR MARKERS

A - ACCEPTED

	MARKER ID	PLIES	EFF %
1	DUA5900215	60.	85.72
2	DUA5900225	85.	84.71
3	DUA5900228	7.	85.87
4	DUA590031	105.	86.92
5	DUA5900234	20.	86.02
6	DUA5900243	5.	85.23
7	DUA5900244	20.	82.91
8	DUA5900248	5.	84.15
9	DUA5900721	55.	88.04
10	DUA5901102	45.	86.89
11	DUA5901308	50.	87.14
12	DUA5901312	5.	87.37

B - REJECTED

DUA5900201 DUA5900206 DUA5900207 DUA5900233
DUA5900236 DUA5900704 DUA5901305

DUMMY MARKERS:

	SIZE	GMTS
1	3230	1.
2	3330	60.
3	3332	70.
4	3528	15.
5	3634	60.
6	3728	95.
7	3730	60.
8	3732	25.
9	3826	30.
10	3932	100.
11	3934	105.
12	4028	5.
13	4030	15.
14	4032	30.
15	4034	90.
16	4132	65.
17	4134	80.
18	4228	25.
19	4230	35.
20	4232	15.
21	4234	50.
22	4328	15.
23	4332	45.
24	4434	20.
25	4632	5.
26	4634	70.

S U M M A R Y :

	COVERAGE GARMTS	COVERAGE %	TOTAL AREA	PATTERN AREA	EFFIC %
REGULAR MARKERS	2209.	65.1	4883.96	4224.81	86.50
DUMMY MARKERS	1186.	34.9	2851.17	2337.96	82.00
T O T A L	3395.	100.0	7735.13	6562.77	84.84

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900225	3232	3230	29.
DUA590031	3434	3432	20.
- " -	3532	3530	25.
DUA5901102	4628	4626	25.
DUA5901308	3430	3428	45.
- " -	3632	3628	25.
- " -	3630	3628	20.
- " -	3830	3826	25.
- " -	4330	4328	10.

FABRIC BALANCE:

WIDTH	AVAILABLE		ASSIGNED TO RM		REMAINS	
IN	YD	SQYD	YD	SQYD	YD	SQYD
58.00	3000.0	4833.3	444.4	716.1	2555.6	4117.3
59.00	4000.0	6555.6	321.5	526.9	3678.5	6028.7
60.00	5000.0	8333.3	2184.6	3641.1	2815.4	4692.3

```
ENTER INPUT DEVICE #
1
 ENTER CONTROL CARD
$S
 COR   : STOP    0000

@
```

```
/RUNLPM2
  REQUIREMENT FILE? OXREQN
  MARKER BANK ? OXMBN
  DUMMY MARKER EFFICIENCY ? .83
  FIRST-BEST INTEGER SOLUTION ? FIRST
 1 REAL AND INT SOLUTIONS
```

MARKAMATIC CUT ORDER PLANNING

76/12/13.   RQ FILE: OXREQN

REGULAR MARKERS:

A - ACCEPTED

	MARKER ID	PLIES	EFF %
1	DUA5900215	60.	85.72
2	DUA5900225	85.	84.71
3	DUA5900231	105.	86.92
4	DUA5900234	20.	86.02
5	DUA5900243	5.	85.23
6	DUA5900248	5.	84.15
7	DUA5900721	55.	88.04
8	DUA5901102	45.	86.89
9	DUA5901308	50.	87.14
10	DUA5901312	5.	87.37

B - REJECTED

DUA5800201 DUA5900206 DUA5900207 DUA5900228
DUA5900236 DUA5900704 DUA5901305

DUMMY MARKERS:

	SIZE	GMTS
1	3230	35.
2	3330	60.
3	3332	70.
4	3528	15.
5	3634	60.
6	3728	95.
7	3730	60.
8	3732	25.
9	3826	30.
10	3928	20.
11	3932	100.
12	3934	105.
13	4028	5.
14	4030	15.
15	4032	30.
16	4034	90.
17	4132	65.
18	4134	80.
19	4228	25.
20	4230	35.
21	4232	15.
22	4234	50.
23	4328	15.
24	4332	45.
25	4434	20.
26	4632	5.
27	4634	70.

S U M M A R Y :

	COVERAGE GARMTS	COVERAGE %	TOTAL AREA	PATTERN AREA	EFFIC %
REGULAR MARKERS	2155.	63.5	4771.14	4131.28	86.59
DUMMY MARKERS	1240.	36.5	2929.52	2431.50	83.00
T O T A L	3395.	100.0	7700.65	6562.77	85.22

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900225	3232	3230	15.
DUA5900231	3434	3432	20.
- " -	3532	3530	25.
DUA5901102	4628	4626	25.
DUA5901308	3430	3428	45.
- " -	3632	3628	25.
- " -	3630	3628	20.
- " -	3830	3826	25.
- " -	4330	4328	10.

FABRIC BALANCE:

WIDTH	AVAILABLE		ASSIGNED TO RM		REMAINS	
IN	YD	SQYD	YD	SQYD	YD	SQYD
58.00	3000.0	4833.3	444.4	716.1	2555.6	4117.3
59.00	4000.0	6555.6	321.5	526.9	3678.5	6028.6
60.00	5000.0	8333.3	2116.9	3528.2	2883.1	4805.2

6.605 CP SECONDS EXECUTION TIME

```
:PR,COR
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$M
CT  OXREQB
  DUMMY MARKER EFFICIENCY :
.83
  ENTER THE AREA OF THE SIZE : 3434
1.8633
ENTER INPUT DEVICE #                          70 seconds
1
  ENTER CONTROL CARD
$E
   FIRST-BEST INTEGER SOLUTION
FIRST
    1   REAL AND INT SOLUTIONS
```

MARKAMATIC CUT ORDER PLANNING

REGULAR MARKERS

A - ACCEPTED

	MARKER ID	PLIES	EFF %
1	DUA5900215	60.	85.72
2	DUA5900225	85.	84.71
3	DUA590031	105.	86.92
4	DUA5900234	20.	86.02
5	DUA5900243	5.	85.23
6	DUA5900248	5.	84.15
7	DUA5900721	55.	88.04
8	DUA5901102	45.	86.89
9	DUA5901308	50.	87.14
10	DUA5901312	5.	87.37

B - REJECTED

DUA5900201 DUA5900206 DUA5900207 DUA5900228
DUA5900233 DUA5900236 DUA5900244 DUA5900704
DUA5901305

DUMMY MARKERS:

	SIZE	GMTS
1	3230	35.
2	3330	60.
3	3332	70.
4	3528	15.
5	3634	60.
6	3728	95.
7	3730	60.
8	3732	25.
9	3826	30.
10	3928	20.
11	3932	100.
12	3934	105.
13	4028	5.
14	4030	15.
15	4032	30.
16	4034	90.
17	4132	65.
18	4134	80.
19	4228	25.
20	4230	35.
21	4232	15.
22	4234	50.
23	4328	15.
24	4332	45.
25	4434	20.
26	4632	5.
27	4634	70.

S U M M A R Y :

	COVERAGE GARMTS	COVERAGE %	TOTAL AREA	PATTERN AREA	EFFIC %
REGULAR MARKERS	2155.	63.5	4771.25	4131.26	86.59
DUMMY MARKERS	1240.	36.5	2929.54	2431.51	83.00
T O T A L	3395.	100.0	7700.79	6562.77	85.22

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900225	3232	3230	15.
DUA590031	3434	3432	20.
- " -	3532	3530	25.
DUA5901102	4628	4626	25.
DUA5901308	3430	3428	45.
- " -	3632	3628	25.
- " -	3630	3628	20.
- " -	3830	3826	25.
- " -	4330	4328	10.

FABRIC BALANCE:

WIDTH	AVAILABLE		ASSIGNED TO RM		REMAINS	
IN	YD	SQYD	YD	SQYD	YD	SQYD
58.00	3000.0	4833.3	444.4	716.1	2555.6	4117.3
59.00	4000.0	6555.6	321.5	526.9	3678.5	6028.7
60.00	5000.0	8333.3	2116.9	3528.2	2883.1	4805.2

```
ENTER INPUT DEVICE #
1
  ENTER CONTROL CARD
$S
  COR    : STOP     0000

@
```

APPLICATION OF LINEAR PROGRAMMING TECHNIQUES IN CUT SCHEDULING FOR BETTER FABRIC UTILIZATION

A preliminary proposal

submitted to

CAMSCO, Inc., 1200 N.Bowser, Richardson, Texas 75080

by

Dr. Milos Konopasek
Associate Professor
School of Textile Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

November 7, 1975

Introduction

The level of fabric utilization in a cutting room depends to a large extent on the way the stock of fabrics (possibly in a few different widths) is scheduled for the products of different styles and sizes. Large bank of markers for different fabric widths and style/size combinations produced on Markamatic system provides a unique basis for optimal scheduling.

The intuitive way of assigning the planned quantities of styles/sizes to available fabrics, starting with the combinations with the lowest waste percentage, usually does not lead to optimum; as a rule, the unfavorable assignments left for the end of the allocation procedure cannot compensate for the favorable ones made at the beginning.

The other possible way of simplified and intuitive solution of the allocation problem, namely projecting the required style/size distribution onto the distribution of style/sizes in specially made markers, may be rather wasteful with respect to both marker-making effort and fabric utilization.

Statement of the problem

Both the exhaustion of the stock and the saturation of the planned quantities of the style/sizes are linear functions of the numbers of layers to be cut using different markers. Therefore the optimum fabric utilization given a certain production plan, fabric stock and marker bank may be solved as a linear programming problem.

The total area of the fabric

$$A = \sum_{j=1}^{m} \sum_{i=1}^{\ell} z_j w_i h_{i,j} \qquad (1)$$

is to be minimized, subject to constraints concerning the production plan

$$\sum_{j=1}^{m} z_j p_{j,k} = s_k \quad \text{for } k=1,2,\dots,n \qquad (2)$$

and the fabric stock available

$$\sum_{j=1}^{m} z_j h_{i,j} \leq t_i \quad \text{for } i=1,2,\dots,\ell \qquad (3)$$

where

w_i is width of the fabric;

t_i is total length of the fabric w_i wide on stock;

z_j is number of layers cut using j-th marker;

$h_{i,j}$ is element of marker length matrix denoting length of j-th marker W_i wide, $h_{\alpha,j} = 0$ for $\alpha = 1, 2, \ldots, i-1, i+1, \ldots, \ell$

$p_{j,k}$ is number of sets of patterns of k-th style/size in j-th marker;

s_k is required quantity of k-th style/size,

all for $i=1,2, \ldots ,\ell$; $j=1,2, \ldots ,m$; $k=1,2, \ldots ,n$,

wkere

ℓ is number of fabric widths,

m is number of markers,

n is number of style/sizes.

It is possible, that the solution vector Z will have to be integer valued and consequently the equal sign in equations (2) - to be replaced by $\geq$. Special care might have to be taken of the remnants at the problem formulation and solution stage as well as in the cutting room.

The technique of generalized linear programming may provide the clue for developing additional markers in order to overcome bottlenecks in the first solutions and to gain an overall waste reduction.

There is also a chance of using one of the less exact but faster and cheaper "transportation problem" algorithms - especially in a case of different fabric widths and homogeneous markers. The possibility of applying this approach in a case of combined markers has to be clarified.

The proposed work

1) Selection the LP algorithm which would fit the purpose within the hardware, computing time and other constraints (main decision to be made between more general simplex method and simpler and faster transportation problem approach; the merits of application of integer programming techniques will also be considered).

2) Design of the data manipulation technique and interactive way of problem setting and resetting, which would fit into the Markamatic system.

3) Writing and debugging relevant FORTRAN programs for HP 2100A computer.

4) Testing the programs on simulated and real problems.

5) Writing up the documentation and draft user's manual.

The research will be finished on or before 7/31 1976; 3 - 4 brief progress reports will be submitted at consultations with the sponsors during January - June 1976.

The sponsors will provide:

1) HP 2100A computer for the period of the work on the research project;
2) information on Markamatic system necessary for fitting the LP segment into it;
3) data for item 4) of the above schedule and access to Markamatic user(s) for the purpose of consulting the feasibility of chosen solutions.

ADDENDUM TO PROPOSAL ENTITLED

"APPLICATION OF LINEAR PROGRAMMING TECHNIQUES IN CUT SCHEDULING FOR BETTER FABRIC UTILIZATION"

DATED NOVEMBER 7, 1975, SUBMITTED BY DR. MILOS KONOPASEK

The Statement of Work, Pages 1 through 5, will be the goal for this programming effort. However, as a minimum, the program provided must be capable of performing the following functions:

Markers generated on the MARKAMATIC System are stored on disk platters. The individual markers are made for a specific combination of styles, material widths, and sizes. In addition, special parameters such as stripe/plaid, nap constraints, cut-downs, etc., are also control factors. When a cutting order consisting of the sizes, styles, width of goods, stripe/plaid constraints, nap constraints, etc., are entered into the system (via cards, TTY, MTU, or floppy disk), the program shall be capable of searching the library of existing markers to select the "best" combination of markers in the library to satisfy the cutting order. The term "best" relates to the results normally obtained when this task is performed manually as compared to the results produced by the program. The program will use linear programming techniques to select the "best" combination of markers. Quite often, markers available in the library will not completely satisfy the cut order request, and, therefore, additional markers will be required.

The final output will, therefore, consist of a list of the markers to be used from the library of markers available plus a list of sizes to be marked to satisfy the cut order.

CUT SCHEDULING FOR OPTIMUM FABRIC UTILIZATION

IN APPAREL PRODUCTION

A THESIS

Presented to

The Faculty of the Division

of Graduate Studies

By

Howard S. Coff

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Textiles

Georgia Institute of Technology

November, 1976

CUT SCHEDULING FOR OPTIMUM FABRIC UTILIZATION

IN APPAREL PRODUCTION

Approved:

Milos Konopasek, Chairman

David R. Gentry

David R. Gentry

John J. Jarvis

Date approved by Chairman: 11/15/76

DEDICATION

I gratefully dedicate this thesis to my parents, Mr. and Mrs. David H. Coff, for their love and encouragement.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my thesis advisor, and friend, Dr. Milos Konopasek, whose guidance, counsel and encouragement made this thesis possible.

I am grateful to Dr. David R. Gentry and Dr. John J. Jarvis for serving on my reading committee.

Special thanks should go to Chris Papaconstadopoulos for his help and guidance in the experimental work on the HP minicomputer.

I would like to express my gratitude to Camsco, Inc. for sponsoring the research project which introduced me to the exciting field of the use of computers for the improvement of fabric utilization in the apparel industry.

I also want to express my thanks to the management of Oxford Industries, Inc. for providing the technical information used in my experiments. Part-time employment with the company during the last stage of my thesis work gave me invaluable insight into the intricacies of practical applications of the results of my research.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SUMMARY

The broad objective of this study is to survey the existing and possible computer assisted techniques to optimize fabric utilization in the apparel plant. Special attention is paid to optimum cut scheduling for given quantities of garments produced from a stock of fabrics using a given variety of markers (pattern layouts). This supplements the existing computerized methods of marker making (layout design) and automatic digitally controlled cutting.

The fabric consumption to be minimized is a linear function of the required quantities of style/sizes and the waste percentages of the applied markers. The constraints concerning the quantities of the garments required and the fabrics available are also linear functions of the marker parameters. Consequently, the optimum cut schedule can be obtained by using linear programming.

An appropriate method of linear programming was selected and a conversational computer program was developed. This method will provide an easy-to-use tool for solving the scheduling problem in every day activities at an apparel manufacturing plant.

CHAPTER I

INTRODUCTION

1.1. Statement of the Problem

Only in the past ten years have there been any new and significant developments in the cutting department. Considering raw material costs typically run 40 percent to 50 percent of the wholesale price of an apparel product, it is difficult to explain this neglect. The level of fabric utilization in a cutting room is determined by three main factors: pattern engineering, marker making, and the selection of markers to cover a particular production plan.*

A basic requirement to achieve high material utilization is designing the patterns in a logical way that brings about an appropriate garment construction consistent with fashion and comfort requirements and also with fabric utilization and garment assembling aspects [1].

The second element which determines fabric utilization is the way in which patterns are arranged on the marker. This is a decisive stage where fabric waste can be minimized; the most progress has been made here in recent years. The most

---

*Making markers is the grouping and interlocking of various sized patterns of a given style and size range into the tightest formations possible within a given width [10].

important advance in this area was an implementation of computer graphics as an aid to a marker designer.

In a typical computer-assisted marker making system, the information on pattern shapes, taken from a digitizer (converts pattern geometry into computer-acceptable numerical form), is stored in the computer memory. Sets of patterns are then displayed on a cathode ray tube (CRT). A marker designer controls the movement of individual panels by a stylus and/or control box to achieve the most efficient layout. The computer prevents overlapping of panels, secures predetermined match of stripes, plaid, etc., stores the information about the markers and relays this information to a computer driven plotter, or, via magnetic tape or disk to a digitally controlled cutting device [5].

Fabric utilization also depends to a large extent on the way the stock of fabrics (possibly in a few different widths) is scheduled for the products of different styles and sizes. In contrast to the computer-assisted marker design, there has been little development in this area. This investigation deals mainly with the third area, optimal marker selection to cover a particular production plan.

The individual markers are made for a specific combination of styles and sizes on a material of a given width. The bank of markers provides a unique basis for selection of relevant markers and optimal scheduling. Quite often existing markers are neglected. This is a waste of previous marker making efforts; it is done because manual retrieval of markers is a cumbersome process.

The selection of markers is now frequently accomplished by either first match or enumeration method. In the first match method, the first combination of markers observed that contain all required style/sizes in the cut order is the one used. This method is obviously inefficient.

The enumeration method is carried out by manually listing several marker combinations in order of efficiency. This is time consuming and the solution is most likely not optimal.

The intuitive way of assigning the planned quantities of styles/sizes to available fabrics, starting with combinations with the highest efficiency percentage, usually will not lead to optimum. The unfavorable assignments left to be allocated at the end cannot compensate for the favorable ones made at the beginning.

The purpose of this study is to survey the existing and possible computerized ways to optimize fabric utilization by selecting the most favorable combination of markers which satisfy the cut order requested. Hopefully, this method will provide an easy-to-use tool for solving the cut scheduling problem in everyday activities at an apparel manufacturing plant.

1.2. Review of Literature

As noted in the previous section, very little has been done or is known about using mathematical methods and computing techniques for optimum scheduling of available fabrics and markers in the cutting department. Therefore, in the survey of literature the emphasis will be on the progress in computer-assisted marker making, because this made efficient cut-scheduling possible and desirable.

The references to the sources of mathematical techniques used during the investigation will be made in Sections 2.1 and 2.3.

Some of the new developments in the cutting department in the last ten years include individual incentive systems, computerized pattern grading,* miniature markers, etc. [5].

Computerized marker making is the bridge between computerized pattern grading and computer controlled cutting. It passes the information from the grading operation to cutting through a computer controlled plotter. The plotter draws full scale master markers from the arrangement of panels created on the CRT and stored in memory. A plotter can draw marks for pockets, darts, and notches. Various statistics, usually the pattern piece area, can be calculated and printed with every graded pattern.

---

*Grading is the process by which a garment pattern--cut to fit a group of persons of a standard size--is made larger and smaller to fit groups of larger and smaller people [1].

Grade rules are how each point (in the pattern) of significance moves inwards and outwards to the corresponding point of other sizes [5]. Grade rules along with digitized patterns are used in both pattern grading and marker making. Therefore, the integrated marker making systems will mean reduced costs in the pattern making area in reference to labor, supplies, and occupancy [3].

The computer-assisted marker making systems primarily reduce the marking labor costs. Productivity increases that of conventional marker making mainly due to the elimination of manually drawing around the patterns which is substituted for by the use of a plotter.

Quality of the marker making operation can also be improved through use of the computerized systems. Pattern pieces can be drawn with improved accuracy because marking restrictions are more easily controlled by the computer software.

The greater productivity provided by the computer-assisted marking systems can be used to meet peak workloads or permit additional effort to improve marker efficiency. If increased capacity is available, then throughput time in marker making can be maintained in periods of peak workloads. The value of any throughput improvement varies depending on what the limiting factors (or binding constraints on the total operation) are for a particular company [4].

One of the most advertised features of the computer assisted marking system is the impact on material utilization. Marking on a reduced scale provides pattern area measurements which when divided by total marker area, gives the marker efficiency. Utilization is improved compared to the conventional method because less time is needed to make the original marker and therefore more time can be used to improve efficiency. Also the fact that "more markers made by your best marker maker using this equipment can reduce fabric waste" is a valid consideration [4].

If every advancement in manual marker making were implemented, the improvement in layout efficiency with computer-assisted systems is probably negligible. But, not every apparel manufacturer has the opportunity to install every innovation in manual marker making. Therefore computer assisted marker making has the advantages in that it can solve the need for rapid response and at a higher efficiency [4].

Computerized pattern grading is the source data of manufacturing control. The data are digitized and controlled by the grading rules to be used. This additional output can be used for further stages in production. Because computerized pattern grading systems perform the translation from a sample size pattern into a full range of sizes, it primarily affects the pattern making and grading labor costs.

Quality is improved by the fact that possible human errors are avoided. Without the accuracies attainable in computerized grading, such things as plaid and stripe cutting and sewing, reduction in seam allowances, and pattern modifications become very tedious.

Computerized grading systems can improve the capacity to process many pattern sets in a given period, thereby reducing cycle time. This is sometimes necessary due to seasonal or style changes. The computerized grading system, like computerized marker making, is most useful to those firms where frequent style changes and short lead times are required. Commercial computer grading services are available for a producer of relatively few styles, who may find it difficult to justify the expense for hardware.

The current state of the art in computerized marker making still requires a man at some point to direct the computer to perform certain functions, such as selection and placement of pattern pieces which are necessary to build the marker [9]. In addition to existing software, some developments have been added to some systems which assist the marker maker. These aids can pack or squeeze the panels in a tighter formation in an attempt to reduce the length once the marker is created. This can improve the marker efficiency achieved by the operator. Additional software has been developed to further reduce waste in creating markers.

There is research currently being done in this area on the methods of arranging the patterns in an optimal formation without human intervention. This usually involves an iterative procedure and/or uses previous experience and predefined marker making rules. An efficient fully automatic marker making system has not yet become a reality.

CHAPTER II

LINEAR PROGRAMMING AND ITS IMPLEMENTATION IN CUT ORDER SCHEDULING

Although some aspects of marker making have been vastly improved, the decision concerning "what to cut from what" is still performed by individuals based on general guidelines. When a cut order (required quantities of garments of given sizes and styles) is received, the first stage of the marking procedure is to decide what combination of sizes/ panels should be arranged together to form a marker. This process is critical because the efficiency is determined here. Many factors such as the number of markers which will have to be created (to satisfy the cut order), the mechanics of laying out the cloth, and the pile height of each marker, must be considered. Because this process is time consuming, whether it is done by retrieving old markers or creating new ones, and is rarely optimal, a different approach was investigated.

2.1. Formulation of Linear Programming Problem

Linear programming deals with the problem of allocating limited resources among competing activities in the best possible way. Linear programming uses a mathematical model to describe the problem. As the name implies, the mathematical functions in this model are required to be linear. Linear programming involves the planning of activities to obtain an optimal result among the feasible alternatives. This can be illustrated by the following simple example. Assume that a company manufactures two products, each with the same raw material costs and each has unlimited demand. Let x_1 and x_2 represent the number of products of product 1 and product 2 produced per hour. The selling price is one dollar for product 1 and two dollars for product 2. X_1 and X_2 are the decision variables for the model and the objective is to choose the combination so as to maximize function Z where

$$Z = X_1 + 2X_2$$

subject to the following restrictions

$$X_1 + X_2 \leq 6$$

$$2X_1 + X_2 \leq 8$$

This problem is of the classic "product mix" type (how much to produce of what). The solution to this problem, solved by program XLINEAR (Appendix A), is to produce 6 units of product two per hour and none of product one.

Both the exhaustion of the stock and the saturation of the planned quantities of the style/sizes are linear functions of the numbers of layers to be cut using different markers. Therefore, the optimum fabric utilization given a certain production plan, fabric stock, and marker bank may be solved as a linear programming problem.

The total area of the fabric

$$A = \sum_{j=1}^{m} \sum_{i=1}^{\ell} Z_j W_i h_{ij}$$

is to be minimized, subject to the constraints concerning the production plan

$$\sum_{j=1}^{m} Z_j P_{j,k} = S_k \qquad \text{for } k = 1,2,\ldots,n$$

and the fabric stock available

$$\sum_{j=1}^{m} Z_j h_{ij} \leq t_i \qquad \text{for } 1,2,\ldots,\ell$$

where

- W_i is width of the fabric
- t_i is total length of the fabric w_i wide on stock
- Z_j is number of layers cut using j-th marker
- $h_{i,j}$ is element of an auxiliary marker length matrix denoting length of j-th marker W_i wide, $h_{\alpha,j} = 0$ for $\alpha = 1,2,\ldots, i-1, i+1,\ldots,\ell$
- $P_{j,k}$ is number of sets of patterns of k-th style/size in j-th marker

S_k is required quantity of k-th style/size,

all for $i = 1, 2, \ldots, \ell$; $j = 1, 2, \ldots, m$; $k = 1, 2, \ldots, n$

where

ℓ is the number of fabric widths

m is number of markers

n is number of style/sizes

Using linear programming, the best possible combination of markers may be selected to satisfy the required sizes as well as several other constraints.

2.2. Integer Linear Programming

The solution of the linear programming problem consists of the combination of markers which give the best overall utilization. The solution will usually include fractional numbers. (For example $x_1 = 2.67$--part of the original solution in Figure 2 (Appendix C). In practice it is meaningless to have 2.67 layers of fabric. Therefore, the solution makes sense only if the decision variables have integer values.

The branch and bound algorithm of pure integer programming is used to restrict the solution to integer values. This method leads to optimum integer solutions through solving a sequence of related non-integer problems.

A typical linear programming problem given by Wagner [12] can be used to illustrate how the integer solution is obtained. The original problem appears as follows:

$$Z = 3x_1 + 3x_2 + 13x_3$$

$$-3x_1 + 6x_2 + 7x_3 \leq 8$$

$$6x_1 - 3x_2 + 7x_3 \leq 8$$

where function z is to be maximized by employing the branch and bound algorithm. The solution to the original problem is $x_1 = x_2 = 2.67$. It must be altered to comply with integer constraints. The tree diagram shows how Wagner obtained the sequence of solutions leading to the integer optimal solution (Figure 1).

The term simplex iteration will refer to a step in the linear programming algorithm which uses the simplex method to reach the optimum. The term branch and bound iteration will be used to indicate an addition of a constraint or "branching" of the problem in seeking to find the optimal integer solution.

The latest branch rule is implemented in order to obtain an integer solution. One advantage of this method is that it decreases the amount of bookkeeping necessary. This rule selects the most recently created subset of branches that has not been fathomed, breaking a tie between subsets created at the same time by taking the one with the most favorable bound (Figure 2).

After some experiments, it was decided to design the

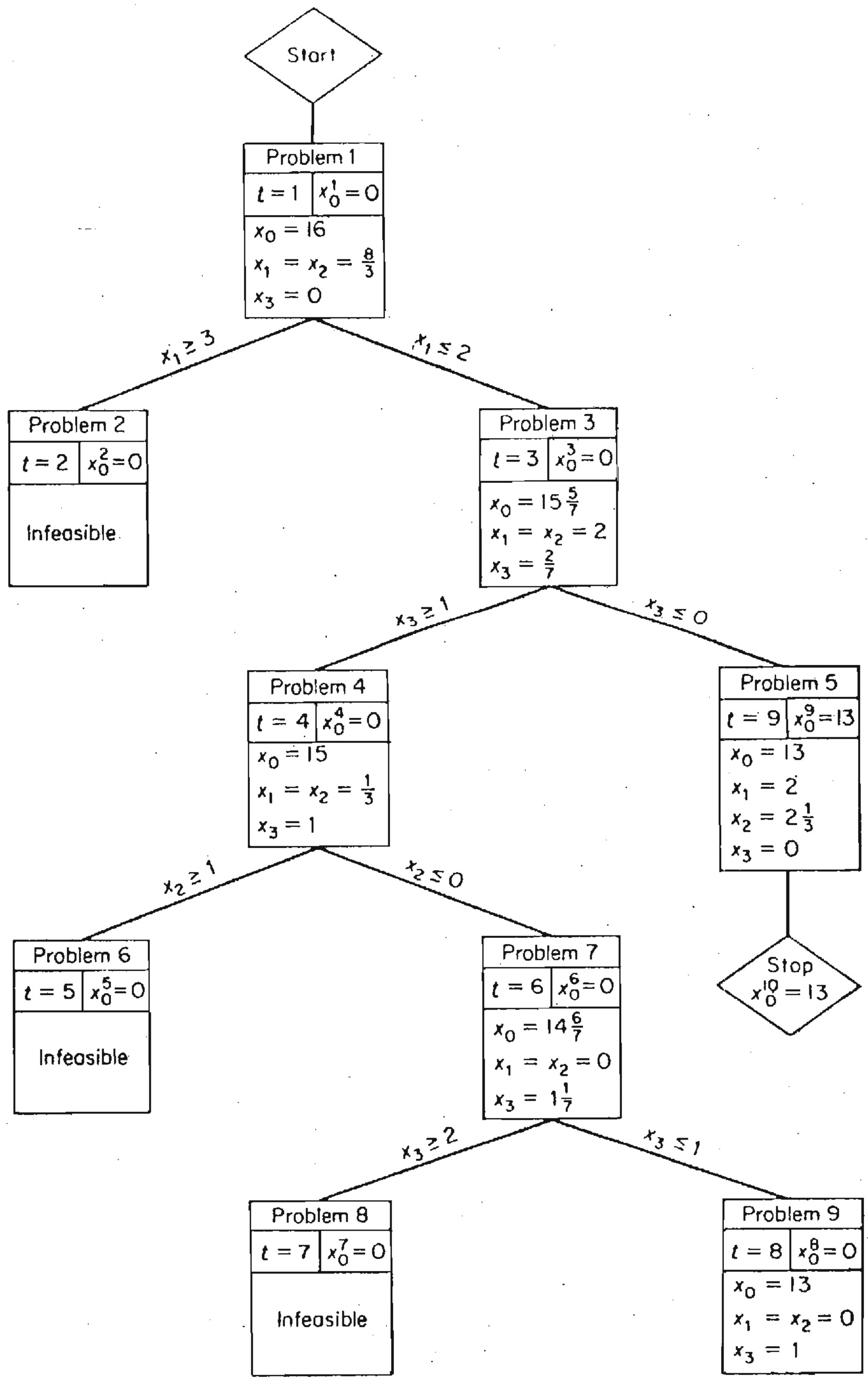


Figure 1. Example of Branch and Bound Method [12]

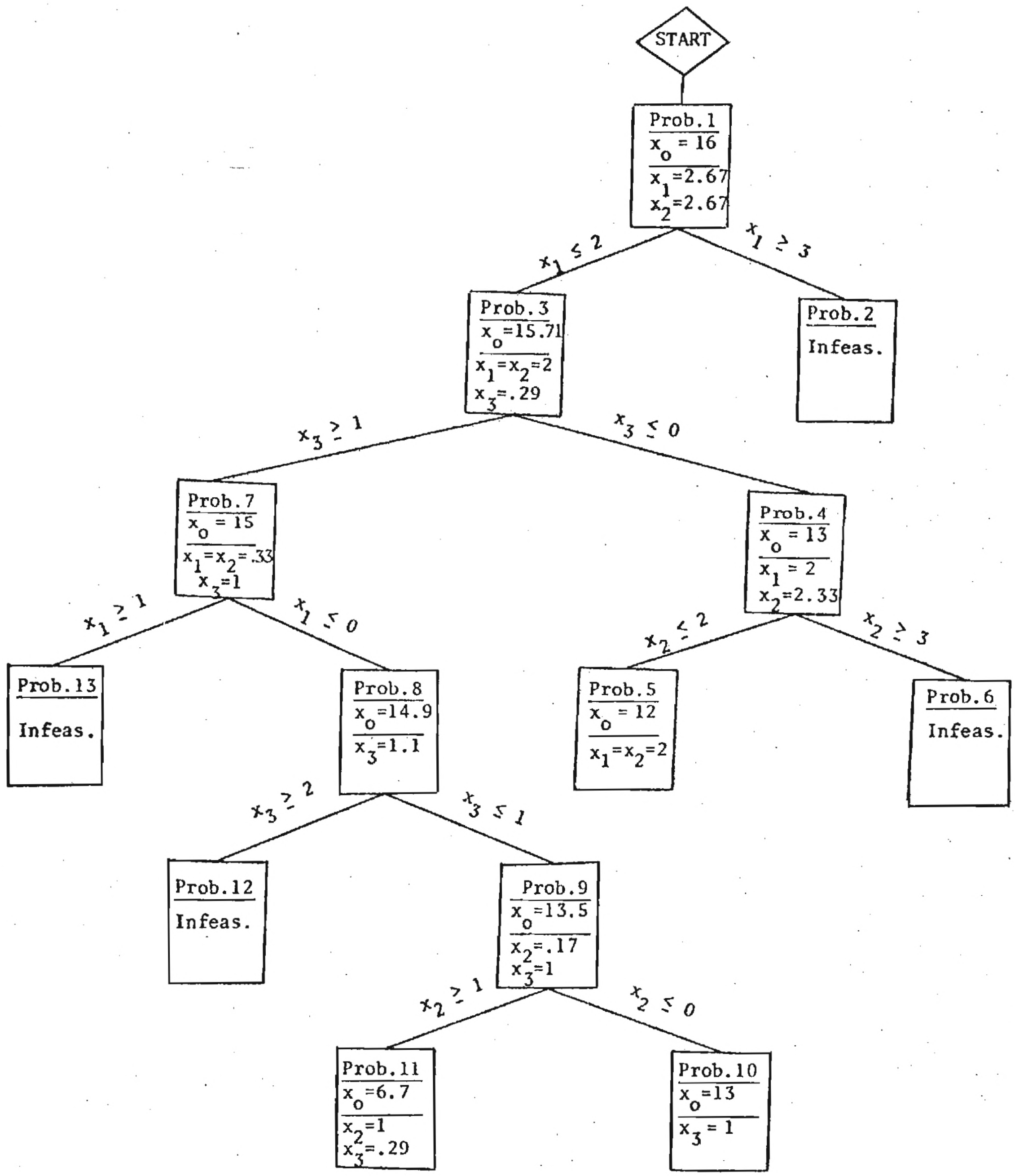


Figure 2. Solution of Integer Programming Problem by Branch and Bound Algorithm as Implemented in LP6

program so that the variable with the smallest positive fractional value is selected to be "split on." This procedure usually allows the program to reach the solution faster. It was also found that if the variable selected is always rounded down first, this will further accelerate convergence to the first integer solution of the cut planning problem.

The original solution found by linear programming is converted to integer by the procedure in Figure 3. Basically the problem begins with the entire set of solutions under consideration. Once it is determined that the solution is feasible, it is further considered. If the solution is integer and it is better (smaller for minimization) than the best integer solution (BIS) up to that point, then this is the new BIS and is stored as the new incumbent solution. If the trial solution (whether it is integer or real) is not better than the present incumbent, the trial solution and potential branches from it is no longer considered. The reason the non-integer solution can be eliminated is because branching further (adding more constraints) can only make the solution value worse. Therefore, at this point a branch of the tree may be fathomed without enumerating all the solutions.

If the solution is not integer, but its value is better than the incumbent, then one of the fractional variables (using the newest bound rule) is "split on."

The following variables and their meanings correspond

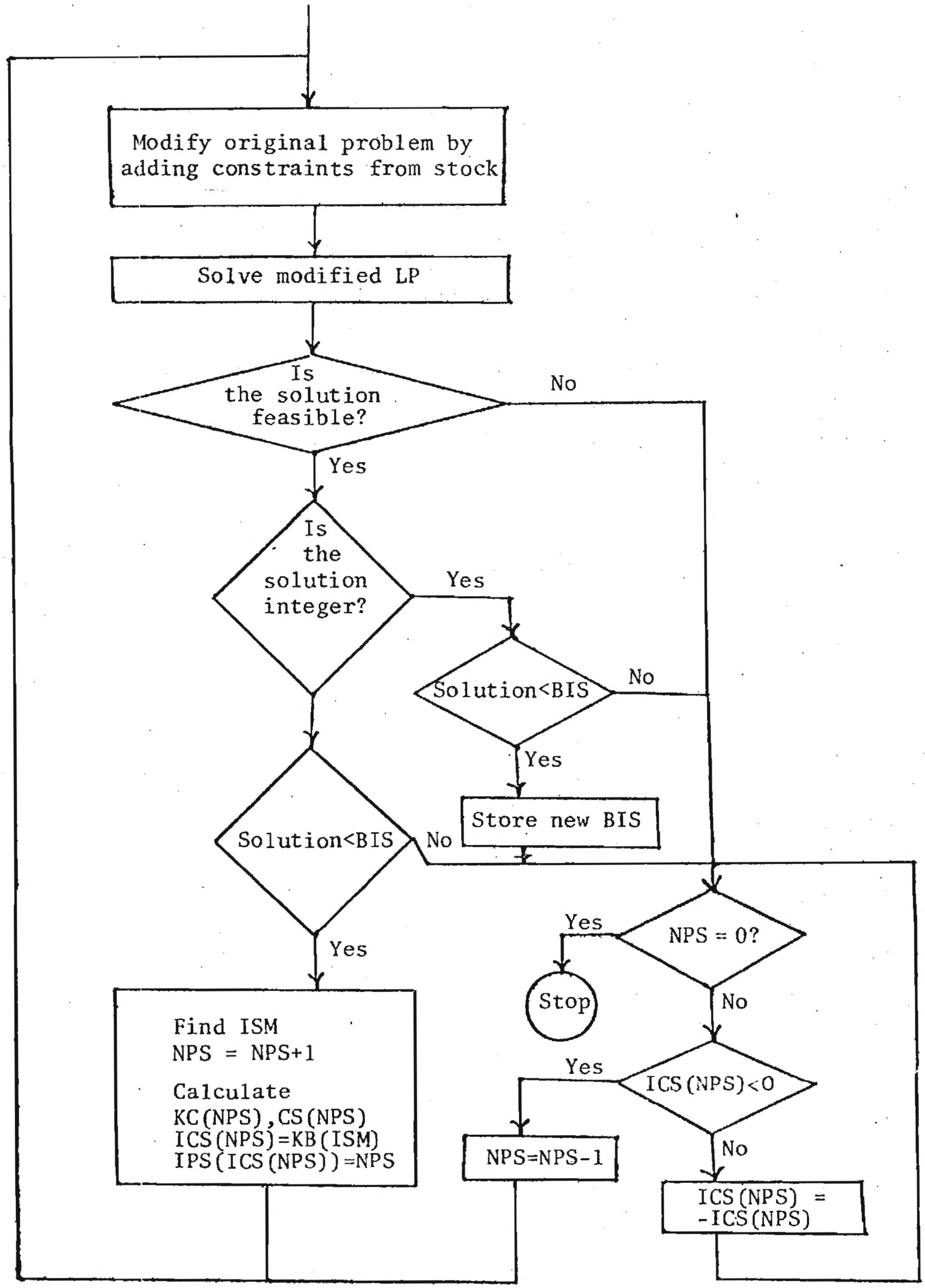


Figure 3. Generating Additional Constraints in Integer Programming

to Figure 3. Note that new variables are chosen in order of increasing positive fractional values as discussed earlier.

BIS	Basic Integer Solution
NPS	Number of additional constraints in stack
ISM	Order index of variable with smallest remainder
KCS(i)	Kind of constraint
CS(i)	Value of constraint
ICS(i)	Index of constrained variable--carrier of branching information
IPS(i)	Pointer to location in constraint stack where constraint of j-th variable is stored
KB	array of pointers linking solution columns with variable indices

The solution of Wagner's example obtained by using this algorithm is shown in Figure 2 and its printout is given in Appendix C. There are some differences between Wagner's tree diagram in Figure 1 and Figure 2; Wagner's solution appears to need fewer branch and bound iteration steps. However, closer examination shows some unexplained short-cuts in Figure 1. For instance, it is difficult to see why after problem 3 the inequality $X_3 \geq 1$ should be preferred to $X_3 \leq 0$ when X_3 is equal to 2/7, i.e., is closer to zero than one. Also, there is no obvious reason why X_2 is constrained in preference to X_1 after problem 4.

Any bounded integer programming problem has a finite number of feasible solutions. However, this finite number usually is very large so that complete enumeration is not practical. Therefore, it is important that only a small fraction of the feasible solutions be examined.

Despite these techniques to reduce the time to implicitly enumerate the solutions, several problems proved to take too long to use this method. This is due to either too many variables and/or a very heavily constrained problem. When dealing with a large problem, requesting only the first integer solution is sometimes necessary due to time limitations. In a large problem the difference between the first integer solution and the optimal solution will usually be marginal. However, in a relatively small problem, the difference may be significant. Because of this situation, the program which prints the solution without the technical data offers the options "first" and "best". When "first" is chosen, the computation is terminated after obtaining and printing the first integer solution. It has been found that this solution is usually optimal or near optimal. In this particular case (Figure 2), the first integer solution value is twelve, whereas the optimal solution value is thirteen (problem 10).

Conditions will determine whether "first" or "best" which gives optimal solution, is appropriate. For instance, if computer time is relatively inexpensive and the cut order is relatively large, it would be worth the extra computation time to request "best" solution. There is also an option ALL in the program LP6 which triggers printing the solution of all non-integer and integer solutions during the Branch and Bound iteration. This option is used in Appendix C. In this case the additional constraints currently in force are printed at the beginning of every branch and bound iteration. Constraints are coded in the following way: the first number is the constraint number; the second number is the subscript of the variable which is constrained. The third number is the kind of constraint (-1 for greater than or equal to and 1 for less than or equal to). The fourth number is the value of a given constraint. The second number is preceded by a negative sign when both possibilities (rounding down and rounding up) for a given variable have been checked out.

For example, problem 2 is constrained by $x_1 \geq 3$ and problem 3 by $x_1 \leq 2$. Both alternatives, by the time problem number three is considered, have been chosen. Therefore the second number is preceded by a negative sign at problem number three.

CHAPTER III

INTERACTIVE LP PROGRAMS

A simple interactive LP (simplex algorithm) program XLINEAR used at Georgia Tech for teaching purposes was selected as a vehicle for preliminary experiments and further development. The problem solving part of the program is satisfactory. However the conversation part is rather poor: the user is expected to type in a complete problem specification (coefficients of object function, constraints, all the matrix coefficients); there is no provision for storing and editing the problem specification and for re-entering the solving part.

A series of modifications have been made in order to improve the conversation part for more efficient experimentation on various aspects of cut scheduling. The modification went through the following stages:

LPØ: changing specified objective function coefficients, constraint values, matrix elements and also repetition of the solution is made possible.

LP1: The code of the solving part was simplified. Interactive alteration of the number of variables and constraints is added. There is a provision made for writing the formulated problem into a data file and accessing stored problems as well as displaying the problem during the conversation.

LP2:	Minor changes and improvement of efficiency.
LP3:	Finalizing the design of conversation. The following responses to computer's request for the next step are available: PROBLEM...initiate entering a problem, either typing in from terminal or reading from data file; DISPLAY...prints objective function coefficients, constraints, and matrix of the current problem.
ALT COF ALT RS ALT MC	alteration of specified objective function coefficient, constraint, or matrix element.
ADD VAR REM VAR ADD CNSTR REM CNSTR	adding or removing any variable or constraint to and from the problem.
STORE	Writes the whole problem into specified data file.
SOLVE	Solves LP problem using simplex method, prints out the trace of iteration (column replacement and current value of object function) and optimal solution.
END	Terminates the conversation.

LP4: Differs from LP3 in the following respects:

(a) Does not require inputting the basic part of the matrix and objective function coefficient. User indicates the type of constraint (+1, 0, -1 for $\leq$, =, $\geq$ respectively) and program creates the basis using the subroutine SUSLAR;

(b) Does not display the base part of the problem;

(c) Prints out the trace of the iteration process only when asked;

(d) Prints the solution in natural order of variables including coefficients of object function for easier interpretation of results (Appendix B).

The flow chart for LP4 is the same as the flowchart for LP6 (see Appendix D) without the "SOLVE INT" option.

LP5: Differs from LP4 in that the internal matrix is represented in a "string" containing only the nonzero coefficients so that less storage is necessary. The number of non zero matrix coefficients increases due to the pivoting procedure in the simplex algorithm so that more numbers must be stored at later iterations. But, even this greater number is considerably less than storing a full matrix. The matrix is stored in three one-dimensional arrays; one storing the actual coefficients; another identifies the row each coefficient is located in; the third adds cumulatively the numbers of nonzero coefficients in each column.

LP6: This is the first version of the program that is capable of reaching an integer solution. A flow chart showing the dialogue in program LP6 is in Appendix D. All the user commands have previously been explained except for "SOLVE INT". This simply tells the computer that an integer solution is desired. Then the computer will ask whether all, only integer, or only best integer solution is to be printed.

LP7: Is identical with LP6 with the only difference in format of the representation of the problem matrix in written and read-in problem data files; whereas the LP6 uses full matrix representation, the LP7 uses a string representation which makes it compatible with files TEMP2 created by the production version of the cut order scheduling program package described in Chapter 4.4.

CHAPTER IV

SPECIAL PROBLEMS IN CUT ORDER PLANNING

The formulation of real linear programming problems in cut order planning involves special considerations which are not covered by the introduction to the problem in Chapter II. This applies in particular to the situations when a given cut order cannot be satisfied by the exclusive use of existing markers, and to so called "cut-downs". It would be possible to account for these and other peculiarities by modifying the linear program manually (using for example interactive program LP6, see Chapter III). However, it would be rather inefficient and time consuming. Therefore in this chapter we concentrate on defining the rules for modifying the linear programming problem so that the necessary manipulation may be programmed and performed automatically by the computer.

4.1. Dummy Markers

Markers available in the marker bank or marker library sometimes cannot completely satisfy the cut order requested. In this case additional markers will have to be made of the remaining sizes to satisfy the cut order.

Utilizing the formulation of the linear programming problem, introduced in Chapter II, the unsatisfied part of the cut order is completed by artificial variables which can come into the solution. These variables are assigned large coefficient values (for a minimization problem) in the objective function and therefore come into the final solution only when the cut order cannot possibly be fulfilled otherwise.

If artificial variables are used there is no way the overall efficiency of the solution can be assessed because the objective function coefficients of artificial variables are worse (higher) than any real marker. All markers regardless of their efficiency, will come into the solution before any artificial variables. The artificial variables will then fill the unsatisfied part of the order; because the objective function coefficients of all the artificial variables are equal and the solution would be biased towards better coverage of smaller sizes by regular markers. This is the reason why the concept of a "dummy marker", that of an imaginary one-size-one garment marker with a given efficiency, is introduced. If the cut order cannot be satisfied at all, the program will automatically bring "dummy markers" into the solution. These dummy markers, one for each style/size, are considered in the problem along with the real markers.

There is one other purpose that these dummy markers serve besides coming into solution when it is impossible for real markers to cover cut order. Dummy markers can be used to allow only the best real markers to come into the solution. For example, the user enters in a desired efficiency, 82 percent. Only markers with 82 percent efficiency or greater will enter into the final solution (with dummy markers present where sizes could not be covered by regular markers). The final output will, therefore, consist of a list of the markers to be used from the library plus a list of sizes that must be marked to satisfy the cut order.

By increasing the desired efficiency fewer regular markers will be able to satisfy the requirement and the number of dummy markers in the solution will increase. On the other hand, if the efficiency asked for is lowered, more markers will be able to comply with this lower efficiency and fewer dummy markers will appear in the solution. It is necessary to make dummy markers competitive with real ones so that they could be added as regular variables in the linear programming problem. The objective function coefficient of each marker is the area of fabric the particular marker requires. Therefore, the area each size/style uses has to be calculated in similar fashion. Oxford Industries, Inc., Monroe, supplied real data for complete style named DUA. Utilizing the markers which had only one style/size, 34 sizes could be formulated by

$$A_p = W \cdot L \cdot E$$

where

A_p is area in square yards utilized

W is width of marker

L is length of marker

E is efficiency percentage for a given marker.

There are, however, a total of 158 different sizes in marker bank DUA. Utilizing the 34 data points already obtained, linear multiple regression was used to obtain the remaining 124 sizes. The two independent variables, length and waist and the corresponding areas were used to formulate the three coefficients a, b, c.

$$A = aW + bL + c$$

Once the three coefficients were found, the areas of every particular style/size could then be calculated. These areas are correct assuming 100 percent efficiency so that by dividing each area by the efficiency required, the objective function coefficient is evaluated.

For instance, if a larger efficiency percentage is wanted, this efficiency will result in smaller values of dummy marker objective function coefficients. Therefore, in a minimization problem the dummy markers have greater probability of appearing in the final solution.

4.2. Cut-Downs

The freedom of selecting the cutting schedule with optimum fabric utilization is usually severely restricted by

(a) fragmentation of orders;

(b) differences between size distribution in the cut order and in the available markers;

(c) difficulties or impossibility to create new markers following exactly the size distribution in the individual orders;

(d) required level of efficiency of running the cutting operations from the viewpoint of labor costs, equipment utilization, etc.

Very often the only way of satisfying (d) under the conditions (a), (b), (c) is to cut the full height of the spread fabric into a number of panels exceeding the order requirements in certain sizes, and consequently to cut down part of the larger panels into smaller ones as required.

This method inevitably leads to a slight increase in waste due to additional reduction of the useful area of some of the panels. However, this may be compensated (even from the viewpoint of material utilization only) by avoiding possible inefficient distribution of sizes in a marker and by considering all the markers available. After all, minimization of fabric consumption takes into account the adverse effect of any additional waste accompanying the cut-downs.

The constraints concerning required number of garments of different sizes, consequently have to be modified in order to reflect the possibility of reducing some of the larger size panels into panels of smaller sizes. No similar problem was found in the literature on linear programming.

The problem may be formulated as follows: The plan requirements for cutting schedule without cut-downs is given by a series of equality constraints (see Chapter II):

$$\sum_{j=1}^{m} Z_j P_{j,k} = S_k \qquad \text{for } k = 1,2,\ldots,n$$

where

Z_j is the j-th component of the solution vector of the number of layers to be cut using j-th marker;

$P_{j,k}$ is number of sets of panels of k-th style/size in j-th marker;

m is number of markers under consideration;

n is number of style/sizes;

S_k is required quantity of k-th style/size

Supposing there is such a group of consecutive style/size items $k = \alpha, \alpha+1, \ldots, \ell, \ldots, \beta-1, \beta$, that panels for every item ℓ may be cut from panels for item $\ell+1$ (but not vice versa). In this case the equality constraints have to be replaced by the following ones:

$$\sum_{k=\alpha}^{\alpha+i} \sum_{j=1}^{m} Z_j P_{j,k} = \sum_{k=\alpha}^{\alpha+i} S_k \qquad \text{for } i = 0,1,\ldots,\beta-\alpha-1$$

$$\sum_{k=\alpha}^{\beta} \sum_{j=1}^{m} Z_j P_{j,k} = \sum_{k=\alpha}^{\beta} S_k$$

For instance, if we have three markers yielding the following quantities of each of four sizes

	M1	M2	M3
S1	2	--	4
S2	--	3	--
S3	2	--	4
S4	1	3	--

and if the required quantities in sizes are

S1	S2	S3	S4
160	200	240	180

and the unknown numbers of layers to be cut from each marker are denoted X_1, X_2, X_3, the constraints allowing cut-downs of larger sizes into smaller are

$$2X_1 + 4X_3 \leq 160$$

$$2X_1 + 3X_2 + 4X_3 \leq 360$$

$$4X_1 + 3X_2 + 8X_3 \leq 600$$

$$5X_1 + 6X_2 + 8X_3 = 780$$

In some situations a two dimensional cut-down is used in order to secure greater flexibility in obtaining a solution. This is used for instance in the manufacturing of slacks where not only the inseam length but also the waist may be reduced. (In the former case the term cut-off is used.)

When allowing for two dimensional cut-downs one has to modify the matrix coefficients and equality constraints in a manner similar to that of one dimensional cut-downs. The values of the matrix coefficients and right hand side values are transformed by cumulation in two directions instead of one:

$$\tilde{X}_{i,j} = \sum_{\zeta=1}^{i} \sum_{\eta=1}^{j} X_{\zeta,n}$$

where

X = original value

$\tilde{X}$ = transformed value

The transformation procedure may be illustrated by the following example. Let the original matrix and constraints be

i,j	Size		A	B	C	D	E	Constraint
1,1	30	27	2	0	0	0	0	= 2
1,2	30	28	0	1	1	0	0	= 4
1,3	30	29	0	0	0	1	0	= 0
1,4	30	30	0	0	0	0	2	= 0
2,1	31	27	0	0	0	1	0	= 0
2,2	31	28	0	0	0	0	5	= 6
2,3	31	29	1	0	0	0	0	= 0
2,4	31	30	0	3	0	0	0	= 1
3,1	32	27	0	0	0	2	0	= 1
3,2	32	28	3	0	0	0	1	= 0
3,3	32	29	0	0	4	0	0	= 3
3,4	32	30	0	0	3	0	0	= 2

The right hand side values are transformed according to the last equation as follows:

	27	28	29	30
30	2	4		
31		6		1
32	1		3	2

Original

	27	28	29	30
30	2	6	6	6
31	2	12	12	13
32	3	13	16	19

Transformed

The matrix columns corresponding to individual markers have to be transformed in a similar way, so that the formulation of the problem will change to the following:

i,j	Size		A	B	C	D	E	
1,1	30	27	2	0	0	0	0	≤ 2
1,2	30	28	2	1	1	0	0	≤ 6
1,3	30	29	2	1	1	1	0	≤ 6
1,4	30	30	2	1	1	1	2	≤ 6
2,1	31	27	2	0	0	1	0	≤ 2
2,2	31	28	2	1	1	1	5	≤ 12
2,3	31	29	3	1	1	2	5	≤ 12
2,4	31	30	3	4	1	2	7	≤ 13
3,1	32	27	2	0	0	3	0	≤ 3
3,2	32	28	5	1	1	3	6	≤ 13
3,3	32	29	6	1	5	4	6	≤ 16
3,4	32	30	6	4	8	4	8	= 19

Note: all the equality constraints but the last are now "less than or equal to."

Only one-dimensional cut-downs are considered in the final version of the program package described in Chapter V.

4.3. Cut-Down Assignment

The modification of the LP problem for cut-downs described in the previous section leads to a solution which satisfies all the requirements and constraints and which may or may not imply cut-downs. The distribution of the cut-downs cannot be seen directly from the solution. The decision about an assignment of cut-downs is a routine and, sometimes, rather laborious task. Performing this task by computer is an obvious choice.

It has been realized that the assignment of cut-downs can be formulated as a separate LP problem. All the possible cut-downs would play the role of unknown variables. The discrepancies between the number of garments required and suggested by the original solution would become right hand side values of the equality constraints. All objective function coefficients would be equal to one and the objective function to be minimized would be the total number of plies to be cut down.

This idea was abandoned after a few experiments. The reason was that from the production viewpoint the number of cut-down cases (bundles) rather than cut-down plies should be minimized. It is not possible to formulate a relevant LP problem and it is not easy to solve this problem by another optimization method and therefore a simple allocation algorithm is used.

The algorithm takes one cut-down group after another and compares the number of garments of each size in the solution with those originally required (i.e. with original constraints before cut-down adjustment). It evaluates the surpluses and deficiencies (if there are any) in each size within a cut-down group. It picks up the largest surplus and allocates it to relevant deficiencies, then the next largest surplus, and so on, until all the surpluses are exhausted and all the deficiencies are made up for (this will always be the case once the solution satisfies all the constraints).

On the following table the cut-down allocations are indicated by single circles (donors) and double circles (recipients). The data are taken from the example discussed in Chapter V and shown in Appendices E, F and G.

Size ID	Required	Dummy Marker	Regular Marker Required	Marker ID, DUA5900... Number of Plies										Regular Marker Solution	Left for Cut-Down
				201	205	206	207	210	213	217	240	243	246		
				16	12	8	3	14	19	18	1	19	4		
2934	19	--	19						19					19	0
3132	14	--	14					14						14	0
3229	22	22	0											0	0
3230	29	23	6							6				0	6
3231	12	--	12							18				18	-6
3234	10	3	7				6				1			7	0
3331	18	--	18							18				18	0
3333	33	33	0											0	0
3427	11	11	0											0	0
3429	19	--	19									19		19	0
3430	25	--	25		6							19		19	6
3431	17	--	17		1	16								16	1
3432	21	--	21	16	12									28	-7
3433	16	--	16		12									16	0
3434	17	17	0											0	0
3628	15	--	15					14			1			15	0
3632	16	--	16	16										16	0
3634	17	17	0											0	0
3729	18	18	0											0	0
3734	23	--	23						19				4	23	0
3930	5	5	0											0	0

CHAPTER V

FINAL SET-UP OF THE PROGRAM PACKAGE

5.1. Cyber 74 Version

The final version of the program package for cut order planning was originally developed and implemented on the Cyber 74 computer. The block diagram of the components of the package and their interaction is given in Figure 4.

The functions of the components are as follows:

LPM2: Master program which serves to interact selection of requirement file and marker bank. It calls the subroutines LPG4 and LP8. After the linear programming problem is generated and solved the LPM2 generates and prints a standard production report.

LPG4: Linear programming problem generator. It reads the information on sizes/style requirements and fabric stock constraints from a selected requirement file, and the information on the characteristics of regular markers and pattern areas from a marker bank. It creates the nonbasic part of the L.P. matrix, objective function coefficients, and right hand side values (including dummy markers). It also adjusts the matrix and RHS values for cut-downs,

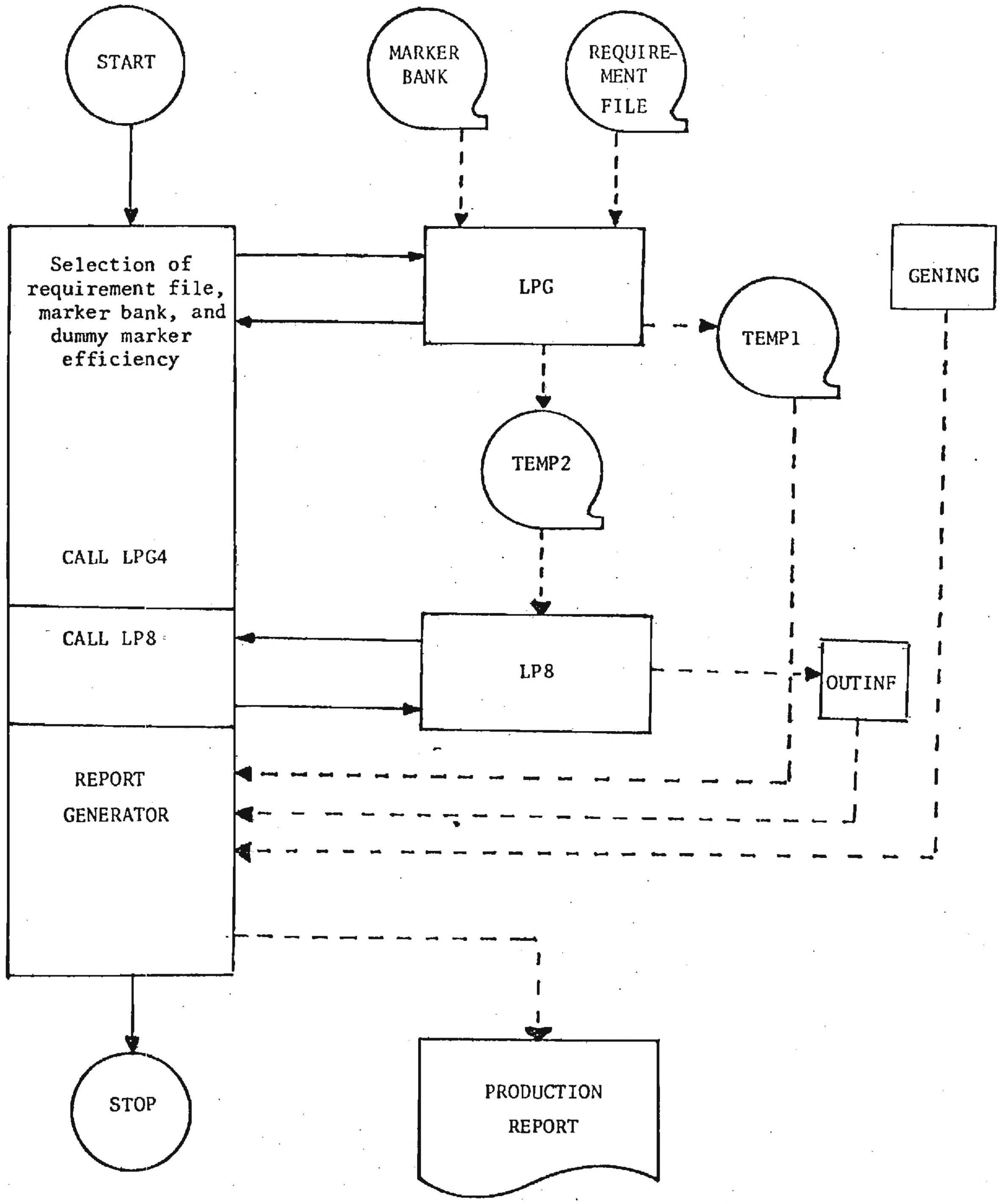


Figure 4. Block Diagram of Cut Order Scheduling Program Package

adds the fabric stock constraints, and writes the LP problem in string representation into the file TEMP2; the original matrix (without dummy markers, cutdowns and stock constraints) is written into the file TEMP1 for future use by LPM2 when generating the production output. More information on marker and size names, etc. for the same purpose is transferred via common block GENINF.

LP8: It is a subroutine built on the basis of the program LP7 discussed in Chapter III. It reads repeatedly the information on the fundamental part of the LP problem from TEMP2 and it solves the series of the real value problems as directed by the branch and bound algorithm (taking into account additional constraints at every branch and bound iteration step. It uses a common block OUTINF for communicating the results of the solution back to the LPM2.

Illustrative examples of input files, temporary files and a production report are given in Appendices E, F and G. The marker bank (Appendix E) consists of two segments. The first contains the values of pattern areas in square yards (second number in each line, i.e. 1.4389, 1.4708, ...) for every size (size identification, first number in each line, i.e. 2926, 2927, ...). This segment is terminated by /. The second segment is divided into subsegments (one for every regular marker) separated by /END. In the first line of each subsegment is the marker identification (e.g. DUA900201), style name (DUA), fabric width in inches and marker length in yards. Each of the following lines of the subsegment gives size identification and the corresponding number of sets of patterns in the marker. The marker bank is terminated by &. The format of the marker bank is temporary and it is supposed to adjust to local conditions of the up-keeping of the marker banks.

The requirement file (Appendix E) consists of the name line (CT...), style identification (ST...), garment requirements (SZ, size identification, number of garments required, optional Y if the cut-down to the previous size is allowed) and fabric stock constraints (PG, width of the fabric in inches, linear yards of the fabric available). The requirement file is terminated by /.

Temporary file TEMP1 consists of three segments: right hand values (constraints), problem matrix coefficients in integer, dense matrix representation, and identifiers of the kind of constraints. The first two segments reflect the situation before adding the dummy markers, cut-down adjustments and adding the fabric stock constraints. In our example we have 21 size-quantity constraints and 13 unknown problem variables (markers, selected as compatible with requirement file). Correspondingly, there are 21 values in both 1st and 3rd segments and 21 x 13 = 273 values in the second segment.

In the temporary file TEMP2 the first line gives the file name and in the second line there are three integers: total number of constraints (in our example 21 for size-quantity requirements and 3 for fabric-widths limits = 24), total number of problem variables (13 regular markers and 21 dummy markers = 34) and number of regular markers (13).
In the next six segments there is full information on the problem: objective function coefficients (34), right hand side or constraint values (24), pointers LQ to the last-row-in-column matrix coefficients (34), string of non-zero matrix coefficients Q (114), string of row indices IQ (114) and identifiers of the type of constraints (24).

The printout of the USER-computer conversation when running the program LPM2, including a production report, is given in Appendix G. The production report is self-explanatory.

5.2. HP Version and Industrial Implementation of the Package

The systems for computer-aided marker making are usually based on and built around a dedicated mini-computer. It is expected that the cut-order planning system will be implemented as an additional function of the marker-making system and it will reside in a mini-computer used for this purpose. One reason for this is that cut order planning will use directly the bank of markers developed and accumulated by the marker-making system.

Therefore the cut-order planning program package developed on Georgia Tech Cyber 74, and described in Section 4.3 of Chapter IV, was transferred to a minicomputer HP21MX. The bulk of the FORTRAN source remains the same although there were changes due to:

(a) differences in Cyber and HP Fortran dialects and word length,

(b) entirely different file writing and reading conventions and procedures

(c) limited storage space on HP

The HP version was divided into 5 modules (Figure 5), controlled by an external monitor, the function of which is to activate (overlay) the proper module depending on the control card (user input).

The communication of the modules is accomplished through permanent disc files, as shown in Figure 6. The information in these files is stored by style and cut identification (user input) on a first in-first out basis. The access to the information related to a particular cut identification (style) is accomplished by supporting directory files.

The above data structure provides the user with a direct access to any of the modulus; therefore repetition of any previous performed operation will require a call only to the module associated with the last phase of the operation.

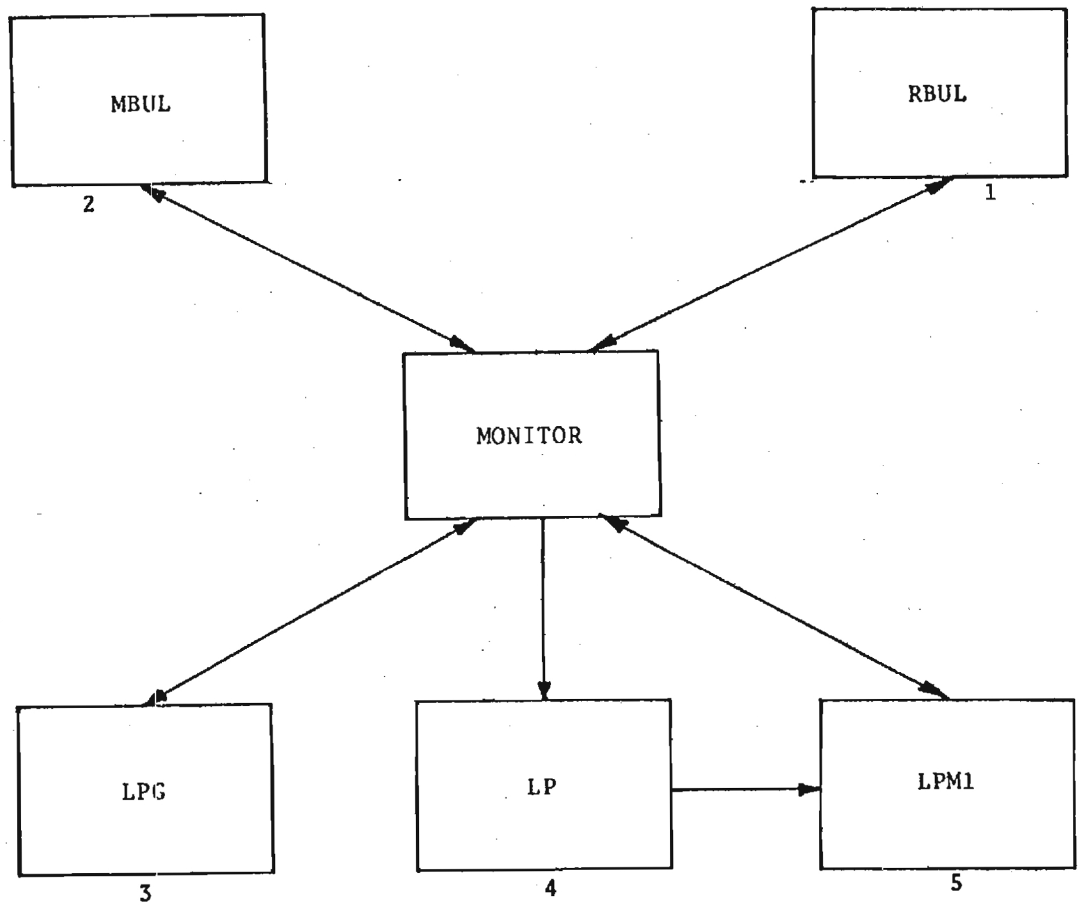


Figure 5. Relationship of the Five Modules of the HP Version of the System

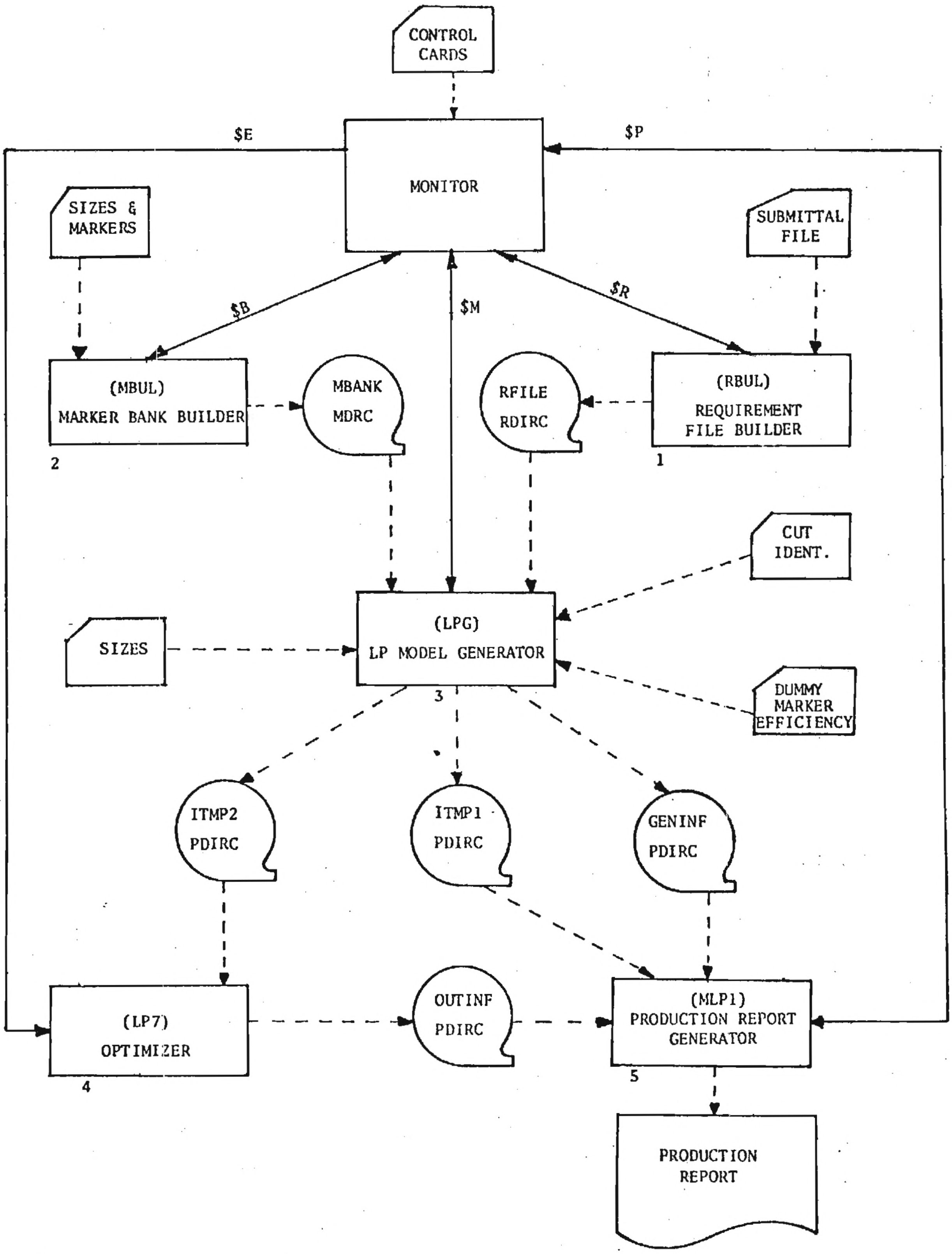


Figure 6. Program Relationship and File Organization of the System

The process is decomposed with respect to time and therefore all the phases can be executed independently. For instance, the user enters a particular submittal file and he generates an LP model with a given dummy marker efficiency. In the sequence the problem is solved and the associated files are updated. The user can resolve the problem for a different dummy marker efficiency by directly accessing module 3 (LPG). The execution of the program goes through the following five phases:

Phase: 1

Program: RBUL--returns control to minitor

Function: Requirement file builder

Control Card: $R

Description: It accepts as input the submitted file and through RDIRC (directory file) it updates the requirement file (RFILE).

Phase: 2

Program: MBUL--Returns control to monitor

Function: Marker bank builder

Control Card: $B

Description: It accepts as input the area sites and the marker specifications and through MDRC (directory file) it updates the marker bank (MBANK).

Phase: 3

Program: LPG--Returns control to monitor

Function: LP model generator

Control Card: $M

Description: LPG operates on RFILE and MBANK. The appropriate markers are selected and introduced as decision variables. LPG interacts with the user to obtain the dummy marker efficiency and any pattern areas missing from MBANK. After the model is built in a matrix form:

(i) dummy markers are introduced as decision variables,

(ii) the module is adjusted by the cut-down specifications,

(iii) fabric stock constraints are added to the model,

(iv) the two-dimensional matrix is converted to a string by a sparse matrix technique.

Finally the model in string form and the supporting tables are stored in permanent files.

Phase: 4

Program: LP8--Passes control to Module 5 (LPM1)

Function: LP Optimizer

Control Card: $E

Description: The optimizer is an Integer Linear Program algorithm which solves the model. The solution and the supporting tables are stored in the permanent file OUTINF.

Phase: 5

Program: LPM1--Returns control to monitor

Function: Production report generator

Control Card: $P

Description: Format specifications for input and control cards.

The following commands are available on the HP version.

1. $REQUEST
2. $BUILD
3. $MODEL
4. $EXECUTE
5. $PRINT

Each of these commands directs the monitor to activate the proper module respectively. Certain data must be entered after the command is accepted.

The following two-character mnemonics are entered in the first two columns.

CT	Cut identification
ST	Style
SZ	Size
PG	Piece goods

The format of the data entered after the commands is shown below. The following conventions are used.

(i) Brackets indicate that the information contained is optional,

(ii) Parentheses indicate that the information contained is a numeric entry,

(iii) Angled brackets indicate that the information contained is a name,

(iv) The character b indicates a blank,

(v) /E terminates the current input stream.

1. $ R[EQUEST]

CT <cut identification>
ST <style name>
SZ <size name>, (quantity)[, Y or, N]

PG (width), (quantity)

/E

2. $ B[UILD]

ST <style name>

<size name>, (area)

<size name>, (area)

.
.
.

/E

<marker name>, (width), (length), [(quantity)*],

<size name>, [(quality)*], <size name>, ... /

<marker name>, ... /

. /@
.
.

/E

3. $ M[ODEL]

CT <cut identification>

4. $ E[XECUTE]

5. $P[RINT]

CHAPTER VI

CONCLUSION

The existing systems for computer-assisted marker making facilitate the development and accumulation, over the period of its operation, of a large number of markers for various garment styles, size distributions, fabric widths, and production conditions.

The preparation of a set of markers for every new cutting order typically includes:

(a) manual selection of those of the old markers which can cover a part of the order;

(b) marking the rest of the order.

The natural limitations of human ability to process large quantities of information impedes manual retrieval and implementation of old markers in new situations. Consequently, the results of past efforts are not fully utilized. The demands on new markers may exceed available operator and system capacity, and the resulting distribution of the order in sizes and quantities, over the set of markers and over the fabric in stock is not as good as it could be.

The developed linear programming system and supporting computer program package for cut order planning performs a large part of the evaluation and decision making which leads to the optimum proportion of (a) and (b) above. The main features of the system are as follows:

--retrieving a marker bank and selecting all the existing markers compatible with a particular cut order;

--formulation and solution of a linear programming problem which gives an optimum composition of the existing markers from the viewpoint of fabric utilization or overall operational costs;

--introduction of a concept of "dummy marker", i.e. an imaginary one-size-one garment marker with assumed efficiency: by varying this efficiency one can cover larger or smaller proportion of the new order by existing markers depending on current priorities;

--provision for cut-downs: system automatically adjusts the required size distribution according to the indicated cut-down options and the output information includes detailed instructions for cut-downs from particular markers and sizes;

--provision for complying with indicated fabric stock constraints: system performs optimum distribution of the order considering the available quantities of the fabrics with different widths;

--flexibility in editing the input information in response to intermediate solutions: if the solution violates some obvious but difficult-to-define limitations, only a part of the solution may be accepted. The original requirements are then adjusted accordingly so that the computer can offer a modified solution, covering the remaining part of the order;

--modular design of the system allows for modification and expansion in compliance with various local conditions and constraints different from those mentioned above.

During the research a series of conversational computing programs for solving linear programming problems has been developed and used. Two of these, LP4 (basic program, dense matrices) and LP6 (sparse matrices, integer programming option) represent a separate by-product of this research and may be easily used for educational and research purposes as shown by Syen [11]. The programs offer all the essential features of interactive computing systems, such as complete freedom and flexibility in manipulating given data (i.e. inserting, altering, or removing variable, constraints, and matrix coefficients.

--Simple means of checking the status of the conversation by displaying current formulation of the problem.

--Means of preventing the corruption of the conversation and basic diagnostics indicating the user's errors and ways of their correction.

--Several levels tracing the iteration procedures for diagnostics and other purposes.

--Simple means of preserving current status of the conversation by writing all the information on the linear programming problem into a data file with an option to recall it whenever necessary.

CHAPTER VII

RECOMMENDATIONS

The results of the research do not solve all the problems of cut order planning as they arise in the real industrial environment. There are two groups of problems future research should be directed towards.

First, there are many additional considerations besides the fabric utilization which in the present cut order planning system may be taken into account only by reformulating and re-entering the problem by the human operator. In the next stage of the research or during the implementation of its results in the industry, some of these considerations should be incorporated in the system so that they would be taken care of automatically. They typically include the following.

The number of markers necessary to satisfy a particular cut order should be minimized for two reasons. Obviously when fewer markers are created, the costs of marking decrease along with the storage requirements in the computer. The second reason the number of markers created should be minimized (or number of garments per marker maximized) is that the larger "bundle size" (garments per marker) usually results in better efficiency. This holds true up to a certain point and then the efficiency levels off.

In conflict with attempting an optimal efficiency in the way stated above, the mechanics of "spreading" or laying out the fabric are such that a minimum of "two-bundles" (two garments per marker) in a spread (total length of fabric on the cutting table) are required.

The pile height, or number of layers of spread fabric, chosen should be as large as possible to reduce the cost in cutting and the table space requirements. This also keeps down the number of markers needed. However, the pile height is restricted by the cutting device used, yarn characteristics (for example textured yarn reduces pile height allowed) and fabric construction (woven or knitted).

Another factor which must be considered is the number of cut-downs allowed for each cut order and the degree to which the patterns are allowed to be cut down. Some plants have their own restriction as to how much a particular panel can be cut down because of the extra waste involved, which is not accounted for in computer assisted marker making and in standard marker characteristics. This aspect is, of course, already considered by the developed program which minimizes the total fabric consumption. However, there is also an additional labor cost involved.

The width of fabric is critical for utilization because some of the patterns closely pack together forming large compound panels or strips; if these cannot be marked on and cut from fabric of compatible width, the cutting waste rises disastrously. On the other hand, ordering, keeping and managing large stock of fabrics with near-to-optimal width distribution is costly and has to be judged on the merits of its potential benefits.

In an ideal situation, the cut order planning system would consider and minimize the total cost of cutting operations and related activities including the cost of the material, rather than to minimize the material consumption only. It is difficult to see how this could be done in the near future. However, at least some of the aspects mentioned above may be incorporated in the present system in the form of additional constraints or penalty functions.

Secondly, the available means of predicting the composition and efficiency of non-existing markers should be incorporated into the system. At some plants and in production of certain apparel products, there is a considerable amount of know-how, accumulated over the period of years, about the ways of combining sizes and styles in a marker. The rules are used for the selection of an efficient composition of styles/sizes entering individual markers, depending on the width of the fabric available, so that the cutting waste does not exceed some given limit. Supposing the rules for the selection of good combinations of sizes for a marker are previously formulated and incorporated in the cut order planning system. In this case the predicted "potential markers" may appear in the linear programming problem along with regular markers and dummy markers. Considering potential markers with trustworthy value of predicted efficiency, rather than filling all the deficiencies by dummy markers, should increase the power of the system and make more competitive the options for cutting the parts of the order, which are not supported by regular markers.

The introduction and utilization of the concept of potential markers will require setting up additional subroutines which will be called from the LP problem generator LPG4 after the information from the requirement file is entered. These subroutines will select the combinations of style/sizes and calculate the expected efficiency according to given rules and a chosen efficiency limit. These subroutines will have to be created individually depending on the local conditions and type of production.

It is envisioned that automatic or semi-automatic marker-making systems could in part play the role of the specialized subroutines. In the future automatic marker-making and cut order planning would merge into one comprehensive system which would accept requirements in terms of pattern shapes and production plan (plus relevant information on additional constraints), and return a complete set of markers together with the production schedule.

APPENDICES

APPENDIX A

SIMPLE EXAMPLE SOLVED BY XLINEAR

```
 XLINEAR
 THIS PROGRAM SOLVES LP PROBLEMS.
 DO YOU DESIRE ADDITIONAL INFORMATION. (YES OR NO)
? NO
 DO YOU HAVE A PROBLEM TO RUN (YES OR NO)
? YES
 ENTER THE NUMBER OF CONSTRAINTS (INTEGER)
? 2
 ENTER THE NUMBER OF VARIABLES (INTEGER)
? 4
 DO YOU DESIRE THAT THE TABLEAU FOR EACH
 ITERATION BE PRINTED (YES OR NO)
? NO
 ENTER THE OBJECTIVE FUNCTION(INCLUDING SLACKS AND ARTIFICIALS)
? 1.,2.,0.,0.
 ENTER THE      2    ROWS OF CONSTRAINTS AS EQUALITIES (DECIMAL)
? 1.,1.,1.,0.,6.
? 2.,1.,0.,1.,8.
0 ITER NO=     0 OBJ FCN=         0.00000
0 REPLACING COL     3  WITH COL     2
0 ITER NO=     1 OBJ FCN=        12.00000
0-----O P T I M A L   S O L U T I O N-----
0ROW  SOLN COL  SOLN VAL    DUAL VAL
    1      2       6.0000     2.0000
    2      4       2.0000     0.0000
       .042 CP SECONDS EXECUTION TIME
```

APPENDIX B

ENTERING, DISPLAYING, SOLVING, STORING, AND TERMINATING THE SIMPLE EXAMPLE PERFORMED BY LP4

```
   PROBLEM  ? NEW
 NO OF CONSTR ? 2
 NO OF VRBLS? 2
 INSERT  2 COEFF OF OBJECT FUNCTION:
? 1,2
 INSERT KIND OF CONSTRAINT AND VALUE:
   1? 1,6
   2? 1,8
 INSERT MATRIX ROW BY ROW,  2 ELEMENTS FOR EACH ROW
 1 ROW  ? 1,1
 2 ROW  ? 2,1

 NEXT ? DISPLAY

   2   2

 OBJECTIVE FUNCTION
  1.0000E+00  2.0000E+00

 M A T R I X

  1ROW:   LE    6.0000
  1.0000E+00  1.0000E+00

  2ROW:   LE    8.0000
  2.0000E+00  1.0000E+00

  NEXT? SOLVE
  PRINT TRACE ? NO
0 ITER NO=  1,OBJ FCN=    -.120000E+02
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL     O F COEFF
   1       2      6.0000    .200E+01       -2.0000
   2       4      2.0000   0.               0.0000

 NEXT ? STORE
 DATA FILE NAME ? EX LP4

 NEXT ? END
      .105 CP SECONDS EXECUTION TIME
```

APPENDIX C

SOLUTION PRINTED BY UTILIZING "ALL" OPTION OF PROGRAM LP6

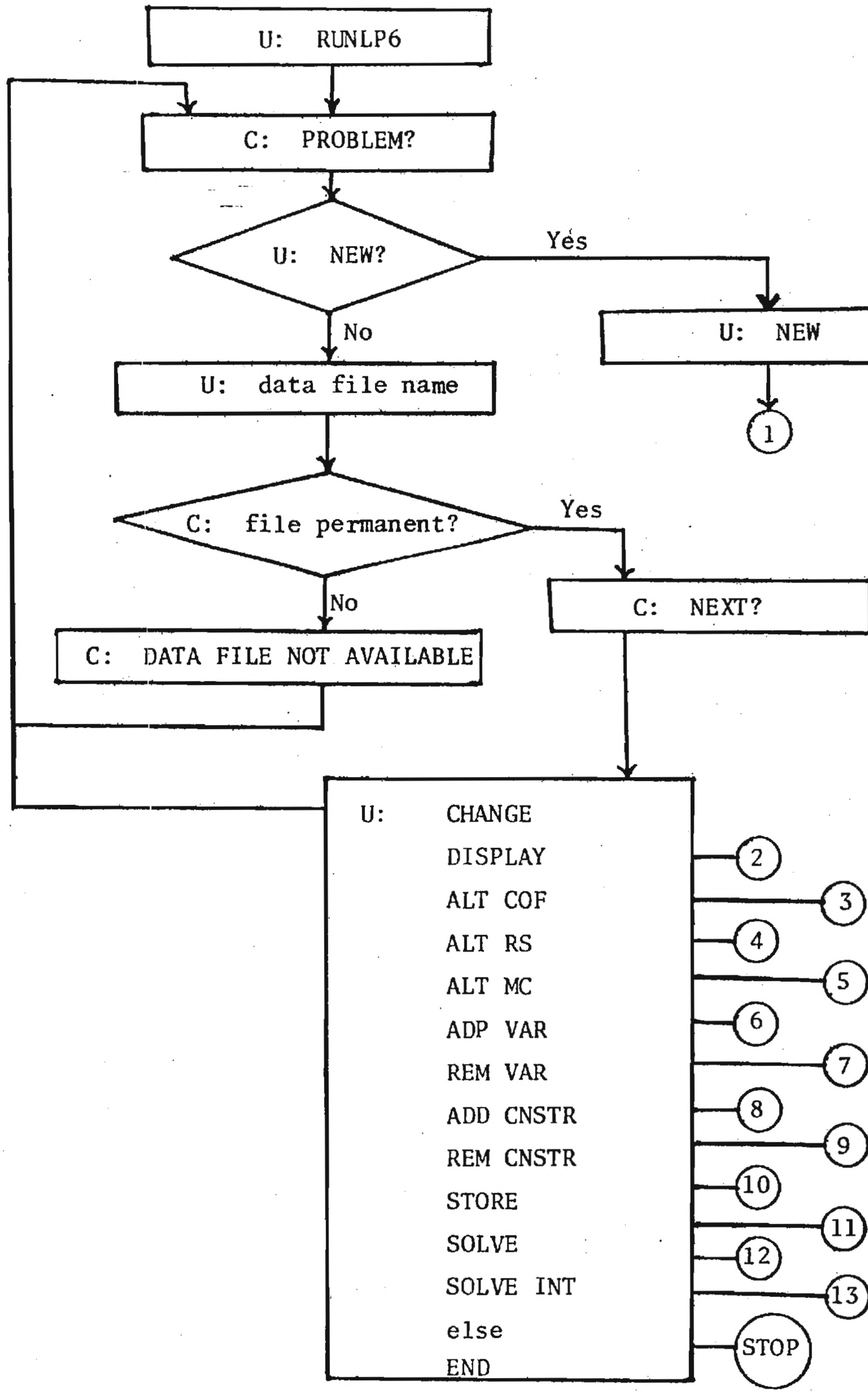
```
/RUNLP6
   PROBLEM  ? BBMAX

 NEXT ? SOLVE INT
 PRINT ALL-INT-BEST ? ALL
  PRINT TRACE ? NO
0 ITER NO=  3,OBJ FCN=    -.160000E+02
0---------O P T I M A L   S O L U T I O N--------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL     O F COEFF
   1        1      2.6667   .100E+01        -3.0000
   2        2      2.6667   .100E+01        -3.0000
 1 1 -1 3.
0 ITER NO=  2,OBJ FCN=     .333317E+06
0---------O P T I M A L   S O L U T I O N--------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL     O F COEFF
   1        1      2.6667   .222E+06        -3.0000
   2        2      2.6667   .111E+06        -3.0000
   3        6       .3333  0.           1000000.0000
 1 -1 1 2.
0 ITER NO=  3,OBJ FCN=    -.157143E+02
0---------O P T I M A L   S O L U T I O N--------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL     O F COEFF
   1        1      2.0000   .905E+00        -3.0000
   2        2      2.0000   .429E+00        -3.0000
   3        3       .2857   .952E+00       -13.0000
 1 -1 1 2.
 2 3 1 0.
0 ITER NO=  4,OBJ FCN=    -.130000E+02
0---------O P T I M A L   S O L U T I O N--------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL     O F COEFF
   1        1      2.0000  0.                -3.0000
   2        2      2.3333   .450E+01        -3.0000
   3        3      0.0000   .950E+01       -13.0000
   4        5      3.0000   .500E+00         0.0000
 1 -1 1 2.
 2 3 1 0.
 3 2 1 2.
0 ITER NO=  4,OBJ FCN=    -.120000E+02
0---------O P T I M A L   S O L U T I O N--------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL     O F COEFF
   1        1      2.0000  0.                -3.0000
   2        2      2.0000   .300E+01        -3.0000
   3        3      0.0000   .130E+02       -13.0000
   4        4      2.0000  0.                 0.0000
   5        5      2.0000   .300E+01         0.0000
```

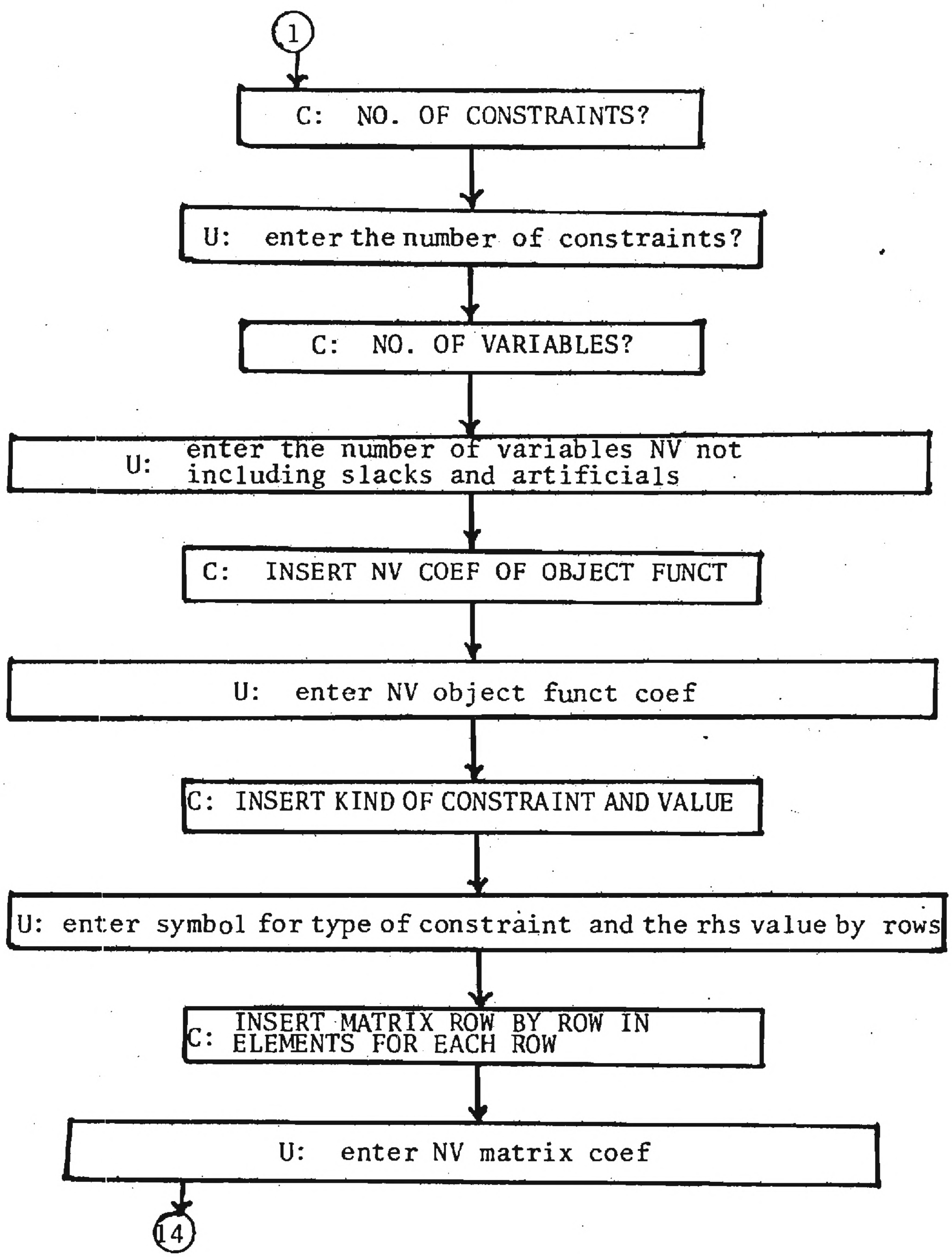
```
 1 -1 1 2.
 2 3 1 0.
 3 -2 -1 3.
0 ITER NO=  2,OBJ FCN=     .666654E+06
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL      O F COEFF
   1        1     2.0000    .500E+06        -3.0000
   2        2     2.3333    .167E+06        -3.0000
   3        5     3.0000   0.                0.0000
   4        7      .6667   0.          1000000.0000
   5        9     0.0000   0.                0.0000
 1 -1 1 2.
 2 -3 -1 1.
0 ITER NO=  4,OBJ FCN=    -.150000E+02
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL      O F COEFF
   1        1      .3333    .100E+01        -3.0000
   2        2      .3333    .100E+01        -3.0000
   3        3     1.0000    .100E+07       -13.0000
   4        6     1.6667   0.                0.0000
 1 -1 1 2.
 2 -3 -1 1.
 3 1 1 0.
0 ITER NO=  4,OBJ FCN=    -.148571E+02
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL      O F COEFF
   1        1     0.0000    .905E+00        -3.0000
   2        2     0.0000    .429E+00        -3.0000
   3        3     1.1429    .100E+07       -13.0000
   4        8      .1429    .952E+00         0.0000
 1 -1 1 2.
 2 -3 -1 1.
 3 1 1 0.
 4 3 1 1.
0 ITER NO=  3,OBJ FCN=    -.135000E+02
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL    DUAL VAL      O F COEFF
   1        1     0.0000    .450E+01        -3.0000
   2        2      .1667    .500E+00        -3.0000
   3        3     1.0000    .950E+01       -13.0000
   4        5     1.5000   0.                0.0000
```

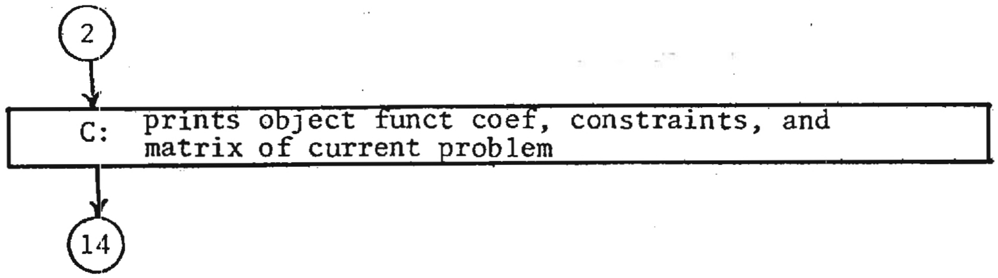
```
 1 -1 1 2.
 2 -3 -1 1.
 3 1 1 0.
 4 3 1 1.
 5 2 1 0.
0 ITER NO=  3,OBJ FCN=    -.130000E+02
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL     DUAL VAL     O F COEFF
   1        1       0.0000    .300E+01       -3.0000
   2        2       0.0000    .300E+01       -3.0000
   3        3       1.0000    .130E+02      -13.0000
   4        4       1.0000   0.                0.0000
   5        5       1.0000   0.                0.0000
 1 -1 1 2.
 2 -3 -1 1.
 3 1 1 0.
 4 3 1 1.
 5 -2 -1 1.
0 ITER NO=  3,OBJ FCN=    -.671429E+01
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL     DUAL VAL     O F COEFF
   1        1       0.0000    .857E+01       -3.0000
   2        2       1.0000    .100E+07       -3.0000
   3        3        .2857    .186E+01      -13.0000
   4        5       9.0000   0.                0.0000
   5        9        .7143   0.                0.0000
 1 -1 1 2.
 2 -3 -1 1.
 3 1 1 0.
 4 -3 -1 2.
0 ITER NO=  2,OBJ FCN=     .857128E+06
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL     DUAL VAL     O F COEFF
   1        1       0.0000    .476E+05       -3.0000
   2        3       1.1429    .952E+05      -13.0000
   3        6       0.0000   0.                0.0000
   4        7        .8571   0.          1000000.0000
 1 -1 1 2.
 2 -3 -1 1.
 3 -1 -1 1.
0 ITER NO=  4,OBJ FCN=     .285699E+06
0---------O P T I M A L   S O L U T I O N---------0
0ROW  VARIABLE  SOLN VAL     DUAL VAL     O F COEFF
   1        1       1.0000    .952E+05       -3.0000
   2        2       1.0000    .476E+05       -3.0000
   3        3        .7143   0.               -13.0000
   4        8        .2857    .571E+06   1000000.0000
 13 REAL AND INT SOLUTIONS
```

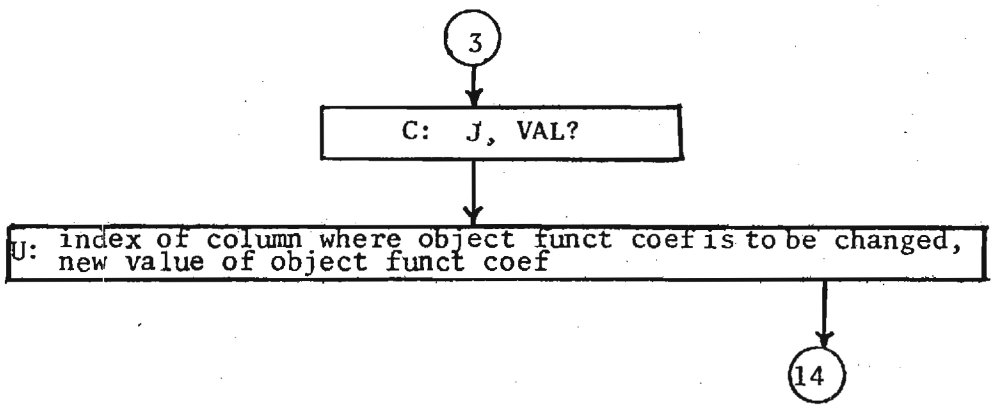
APPENDIX D

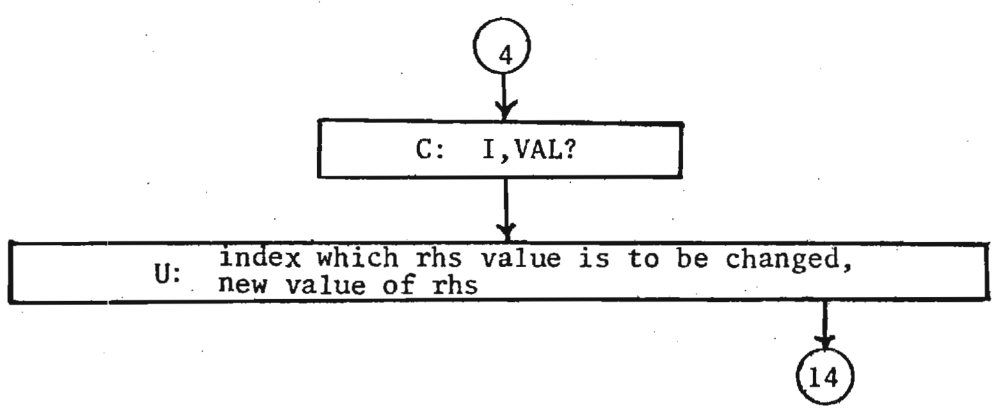
LP6 FLOWCHART

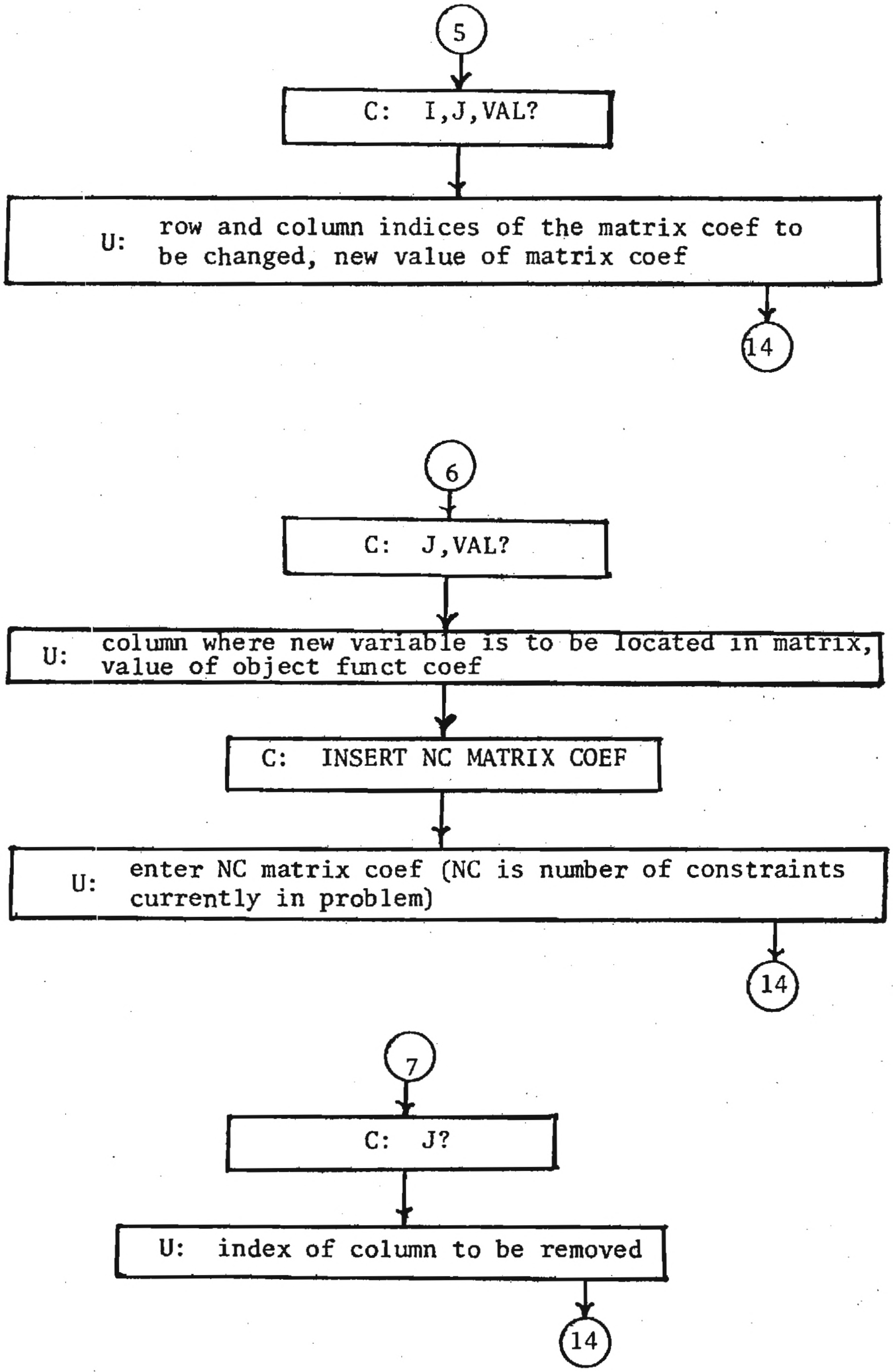

U: RUNLP6
C: PROBLEM?
U: NEW?
Yes
U: NEW
1
No
U: data file name
C: file permanent?
Yes
C: NEXT?
No
C: DATA FILE NOT AVAILABLE
U: CHANGE
DISPLAY
ALT COF
ALT RS
ALT MC
ADP VAR
REM VAR
ADD CNSTR
REM CNSTR
STORE
SOLVE
SOLVE INT
else
END
2
3
4
5
6
7
8
9
10
11
12
13
STOP

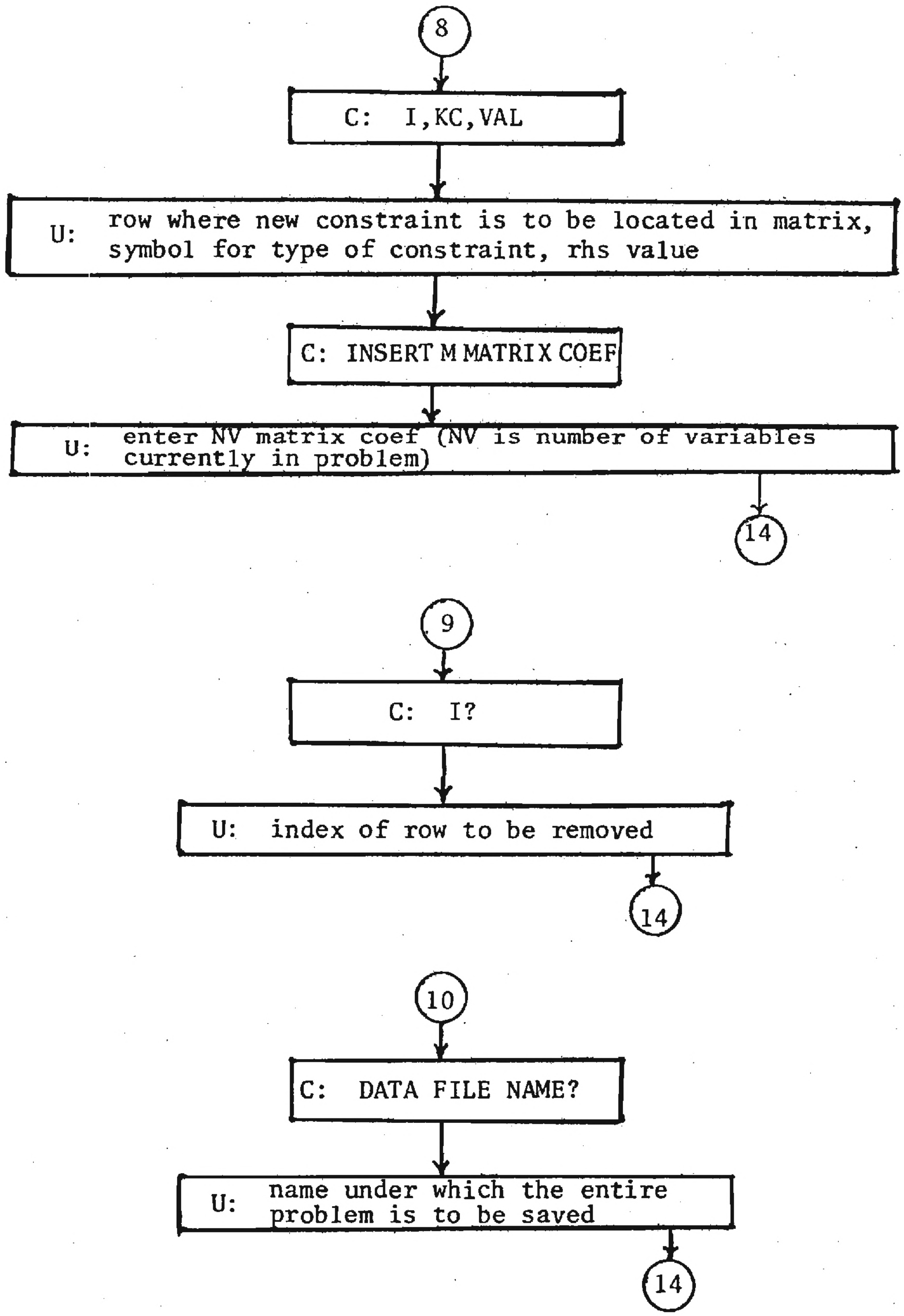

1
C: NO. OF CONSTRAINTS?
U: enter the number of constraints?
C: NO. OF VARIABLES?
U: enter the number of variables NV not including slacks and artificials
C: INSERT NV COEF OF OBJECT FUNCT
U: enter NV object funct coef
C: INSERT KIND OF CONSTRAINT AND VALUE
U: enter symbol for type of constraint and the rhs value by rows
C: INSERT MATRIX ROW BY ROW IN ELEMENTS FOR EACH ROW
U: enter NV matrix coef
14

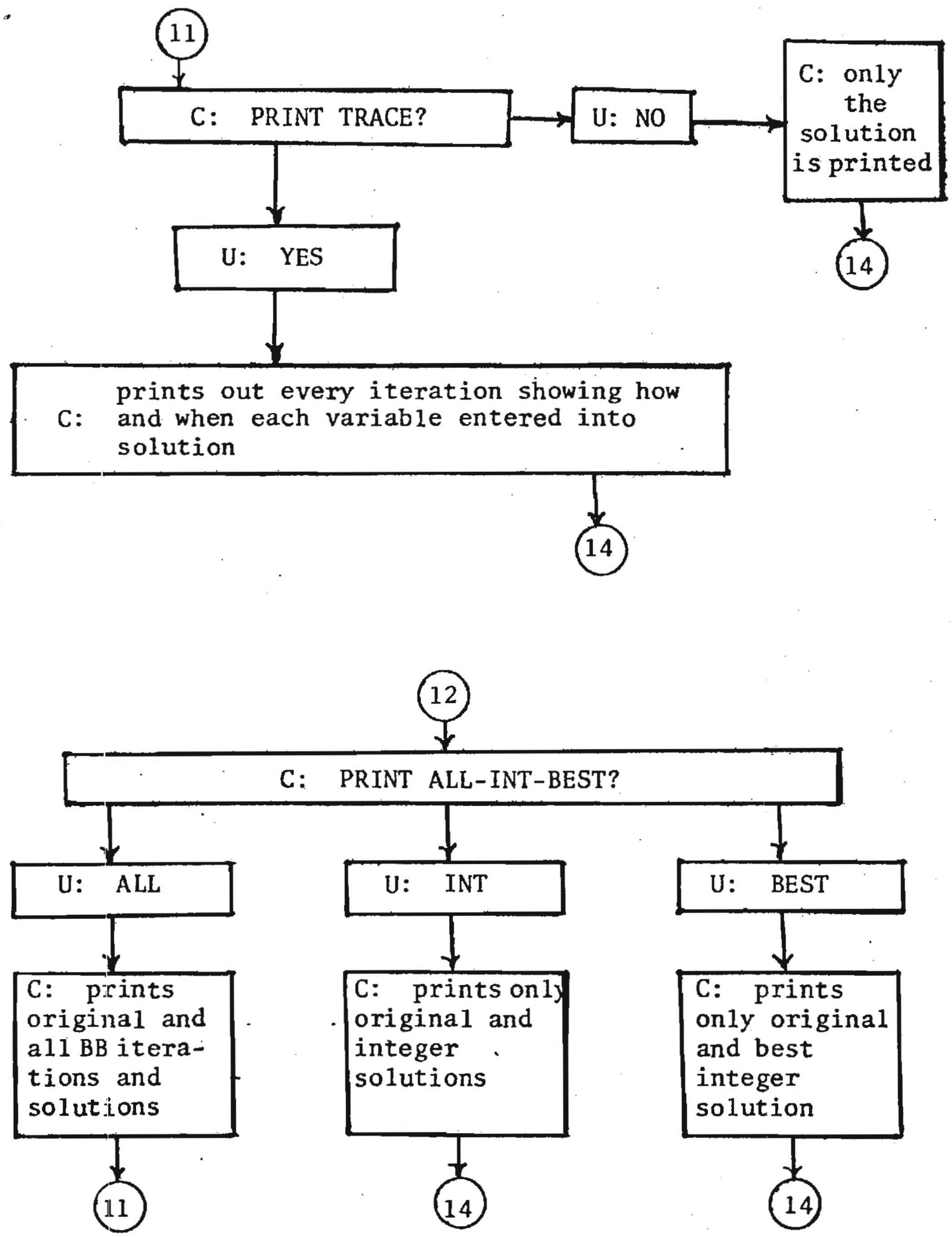

2
C: prints object funct coef, constraints, and matrix of current problem
14


3
C: J, VAL?
U: index of column where object funct coef is to be changed, new value of object funct coef
14

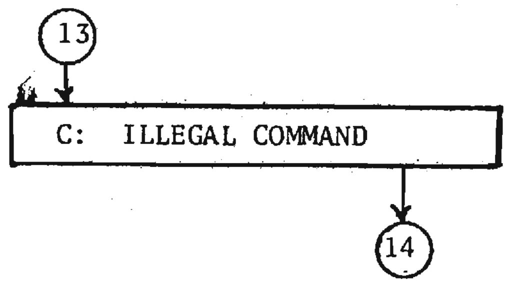

4
C: I,VAL?
U: index which rhs value is to be changed, new value of rhs
14


5
C: I,J,VAL?
U: row and column indices of the matrix coef to be changed, new value of matrix coef
14
6
C: J,VAL?
U: column where new variable is to be located in matrix, value of object funct coef
C: INSERT NC MATRIX COEF
U: enter NC matrix coef (NC is number of constraints currently in problem)
14
7
C: J?
U: index of column to be removed
14


8
C: I,KC,VAL
U: row where new constraint is to be located in matrix, symbol for type of constraint, rhs value
C: INSERT M MATRIX COEF
U: enter NV matrix coef (NV is number of variables currently in problem)
14
9
C: I?
U: index of row to be removed
14
10
C: DATA FILE NAME?
U: name under which the entire problem is to be saved
14



11
C: PRINT TRACE?
U: NO
C: only the solution is printed
14
U: YES
C: prints out every iteration showing how and when each variable entered into solution
14
12
C: PRINT ALL-INT-BEST?
U: ALL
U: INT
U: BEST
C: prints original and all BB iterations and solutions
C: prints only original and integer solutions
C: prints only original and best integer solution
11
14
14


13
C: ILLEGAL COMMAND
14

APPENDIX E

INPUT FILES

2926,1.4389
2927,1.4708
2928,1.5026
2929,1.5344
2930,1.5663
2931,1.5981
2932,1.6300
2933,1.6618
2934,1.6937
3026,1.4728
3027,1.5047
3028,1.5365
3029,1.5684
3030,1.6002
3031,1.6321
3032,1.6639
3033,1.6957
3034,1.7276
3126,1.5068
3127,1.5386
3128,1.5704
3129,1.6023
3130,1.6341
3131,1.6660
3132,1.6978
3133,1.7297
3134,1.7615
3226,1.5407
3227,1.5725
3228,1.6044
3229,1.6362
3230,1.6681
3231,1.6999
3232,1.7317
3233,1.7636
3234,1.7954
3326,1.5746
3327,1.6065
3328,1.6383
3329,1.6701
3330,1.7020
3331,1.7338
3332,1.7657
3333,1.7975
3334,1.8294
3426,1.6085
3427,1.6404
3428,1.6722
3429,1.7041
3430,1.7359
3431,1.7678
3432,1.7996
3433,1.8314
3434,1.8633
3526,1.6425
3527,1.6743
3528,1.7061
3529,1.7380
3530,1.7698
3531,1.8017
3532,1.8335
3533,1.8654
3534,1.8972
3626,1.6764
3627,1.7082
3628,1.7401
3629,1.7719
3630,1.8038
3631,1.8356
3632,1.8674
3633,1.8993
3634,1.9311
3726,1.7103
3727,1.7422
3728,1.7740
3729,1.8058
3730,1.8377
3731,1.8695
3732,1.9014
3733,1.9332
3734,1.9651
3826,1.7442
3827,1.7761
3828,1.8079
3829,1.8398
3830,1.8716
3831,1.9035
3832,1.9353
3833,1.9671
3834,1.9990
3926,1.7782
3927,1.8100
3928,1.8418
3929,1.8737
3930,1.9055
3931,1.9374
3932,1.9692
3933,2.0011
3934,2.0329

```
/END
DUA5900215,DUA,59,2.52
3430,1
3432,1
/END
DUA5900216,DUA,59,2.61
3234,1
3534,1
/END
DUA5900217,DUA,59,2.43
3231,1
3331,1
/END
DUA5900232,DUA,59,2.54
3132,1
3333,1
/END
DUA5900233,DUA,60,2.64
3330,1
3931,1
/END
DUA5900234,DUA,60,2.53
3629,1
3732,1
/END
DUA5900235,DUA,60,2.58
3229,1
3634,1
/END
DUA5900240,DUA,60,2.54
3234,1
3628,1
/END
DUA5900241,DUA,60,2.45
3432,1
3627,1
/END
DUA5900242,DUA,60,2.43
3530,2
/END
DUA5900243,DUA,60,2.44
3429,1
3430,1
/END
DUA5900244,DUA,60,2.53
3230,1
3929,1
/END
DUA5900245,DUA,60,2.72
3532,1
3534,1
/END
DUA5900246,DUA,60,2.67
3433,1
3734,1
/END
&
```

```
DUA5900201,DUA,58,2.55
3432,1
3632,1
/END
DUA5900202,DUA,58,2.36
3329,1
3627,1
/END
DUA5900203,DUA,58,3.08
3633,1
3828,1
/END
DUA5900204,DUA,58,2.37
3030,1
3329,1
/END
DUA5900205,DUA,58,2.59
3432,1
3433,1
/END
DUA5900206,DUA,58,2.52
3431,2
/END
DUA5900207,DUA,58,2.65
3234,2
/END
DUA5900208,DUA,58,2.62
3234,1
3528,1
/END
DUA5900209,DUA,58,2.62
3234,1
3528,1
/END
DUA5900210,DUA,59,2.45
3132,1
3628,1
/END
DUA5900211,DUA,59,2.64
3533,1
3829,1
/END
DUA5900212,DUA,59,2.59
3334,1
3626,1
/END
DUA5900213,DUA,59,2.63
2934,1
3734,1
/END
DUA5900214,DUA,59,2.50
3627,2
```

```
CT OXREQT
ST DUA
SZ 2934,19
SZ 3132,14
SZ 3229,22
SZ 3230,29,Y
SZ 3231,12,Y
SZ 3234,10,Y
SZ 3331,18
SZ 3333,33,Y
SZ 3427,11
SZ 3429,19,Y
SZ 3430,25,Y
SZ 3431,17,Y
SZ 3432,21,Y
SZ 3433,16,Y
SZ 3434,17,Y
SZ 3628,15
SZ 3632,16,Y
SZ 3634,17,Y
SZ 3729,18
SZ 3734,23,Y
SZ 3930,5
PG 58,100
PG 59,150
PG 60,120
/
```

APPENDIX F

TEMPORARY FILES

TEMP1

```
1.9000E+01  1.4000E+01  2.2000E+01  2.9000E+01  1.2000E+01  1.0000E+01
1.8000E+01  3.3000E+01  1.1000E+01  1.9000E+01  2.5000E+01  1.7000E+01
2.1000E+01  1.6000E+01  1.7000E+01  1.5000E+01  1.6000E+01  1.7000E+01
1.8000E+01  2.3000E+01  5.0000E+00

 0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    1    0
 0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1    0
 0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    2
 0    0    0    0    0    0    0    0    0    0    0    0    0    0    2    0    0    0
 0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0
 0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    1    0    0
 0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0
 0    0    0    0    0    0    0    0    0    0    1    0    1    0    0    0    0    0
 0    0    0    0    0    0    0    1    0    1    0    0    0    0    0    0    0    0
 0    0    0    0    0    0    0    1    0    0    0    0    0    1    0    0    0    0
 0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0
 0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    1
 0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
 0    0    0    0    0    0    1    1    0    0    0    0    0    0    0    0    0    0
 0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0
 0    1    0

 0    0    1    1    1    0    1    0    1    1    1    1    1    1    0    1    1    0
 1    0    0
```

Page 79 missing from report.

APPENDIX G

PRODUCTION REPORT

```
RUNLPM2
  REQUIREMENT FILE? OXREQT
  MARKER BANK ? OXMBSS
  DUMMY MARKER EFFICIENCY ? :83
  FIRST-BEST INTEGER SOLUTION ? FIRST
 3 REAL AND INT SOLUTIONS

                    MARKAMATIC CUT ORDER PLANNING
                    -----------------------------
                    76/11/19:       RQ FILE: OXREQT

  REGULAR MARKERS:

    A - ACCEPTED
                    MARKER ID       PLIES
           --------------------------------
           1        DUA5900201       16:
           2        DUA5900205       12:
           3        DUA5900206        8:
           4        DUA5900207        3:
           5        DUA5900210       14:
           6        DUA5900213       19:
           7        DUA5900217       18:
           8        DUA5900240        1:
           9        DUA5900243       19:
          10        DUA5900246        4:

    B - REJECTED
                    DUA5900215   DUA5900232   DUA5900235

 DUMMY MARKERS:
                    SIZE            GMTS
           --------------------------------
           1        3229             22:
           2        3230             23:
           3        3234              3:
           4        3333             33:
           5        3427             11:
           6        3434             17:
           7        3634             17:
           8        3729             18:
           9        3930              5:
```

S U M M A R Y :

	COVERAGE GARMTS	COVERAGE %	TOTAL AREA	PATTERN AREA	EFFIC %
REGULAR MARKERS	228.	60.5	470.19	404.23	85.97
DUMMY MARKERS	149.	39.5	317.65	263.65	83.00
T O T A L	377.	100.0	787.84	667.88	84.77

CUT-DOWNS:

MARKER ID	FROM SIZE	TO SIZE	PLIES
DUA5900205	3432	3430	6.
- " -	3432	3431	1.
DUA5900217	3231	3230	6.

FABRIC BALANCE:

WIDTH IN	AVAILABLE YD	AVAILABLE SQYD	ASSIGNED TO RM YD	ASSIGNED TO RM SQYD	REMAINS YD	REMAINS SQYD
58.00	100.0	161.1	100.0	161.1	.0	.0
59.00	150.0	245.8	128.0	209.8	22.0	36.0
60.00	120.0	200.0	59.6	99.3	60.4	100.7

4.522 CP SECONDS EXECUTION TIME

BIBLIOGRAPHY

1. Abrahams, L., Computer Pattern Grading Techniques as Compared to Manual Grading, Proceedings of a Hatra Conference held in Nottingham, England, April, 1974.

2. Bazaraa, M. S. and J. J. Jarvis, Linear Programming and Network Flows, School of Industrial and Systems Engineering, Georgia Institute of Technology, 1974.

3. Camsco: Report to Industry, Richardson, Texas, 1976.

4. Eberly, N. D. and J. E. Troutman, New Technologies for the Cutting Department, AAMA Computer Assisted Apparel Production/Distribution Conference, October, 1975.

5. Gaetan, M., Computerized Marker Making, Bobbin, March, 1973.

6. Gaetan, M., A Trained Computer Makes Markers Automatically, Bobbin, January, 1974.

7. Garfinkel, R. S. and G. L. Nemhauser, Integer Programming, John Wiley & Sons, 1972.

8. Geoffrion, A. M., Integer Programming by Implicit Enumeration and Balas Method, SIAM Review, Volume 9, No. 2, April, 1967.

9. Hrnack, D. M., Single-Ply Cutting Systems, AAMA Computer Assisted Apparel Production/Distribution Conference, October, 1975.

10. Silverman, M., Marker-Making Manual, Michael Silverman, Hull, Mass., 1964.

11. Syen, S., M. S. Thesis, Georgia Institute of Technology, Atlanta, Georgia, 1976.

12. Wagner, H. M., Principles of Operation Research, Prentice-Hall, Englewood Cliffs, N. J., 1975.